

GE Fanuc Automation

Speicherprogrammierbare Steuerungen

SPS Series 90-30/20/Micro
CPU-Befehlssatz

Referenzhandbuch

GFK-0467L-GE *Junii1999*

GFL-002

In diesem Handbuch verwendete Warn-, Vorsichts- und allgemeine Hinweise

Warnung

Durch Warnhinweise wird in diesem Handbuch hervorgehoben, dass in dem entsprechenden Gerät oder bei dessen Benutzung gefährliche Spannungen, Stromstärken, Temperaturen oder andere Bedingungen bestehen, die zu körperlichen Schäden führen können.

Solche Warnhinweise werden für Situationen verwendet, in denen es durch Unaufmerksamkeit zu Personen- oder Sachschäden kommen kann.

Vorsicht

Vorsichtshinweise werden dann verwendet, wenn es durch mangelnde Sorgfalt zu Schäden an den Geräten kommen kann.

Hinweis

Hinweise dienen ausschließlich dazu, Informationen hervorzuheben, die für das Verständnis und den Betrieb der Geräte von Bedeutung sind.

Dieses Handbuch beschreibt Systembetrieb, Fehlerbehandlung und Logicmaster 90™ Programmieranweisungen für die speicherprogrammierbaren Steuerungen der Series 90™-30, Series 90-20 und Series 90 Micro. Die speicherprogrammierbaren Steuerungen der Series 90-30, Series 90-20 und Series 90 Micro gehören zur Produktfamilie der SPS Series 90 von GE Fanuc Automation.

Revisionen in diesem Handbuch

- Die CPU Modell 364 (Ausgabestand 9.0 und höher) unterstützt den Anschluss an ein Ethernet-Netzwerk über einen oder zwei integrierte Ethernet-Ports. Es stehen AAUI- und 10BaseT-Ports zur Verfügung. Die CPU Modell 364 ist die einzige CPU der Series 90-30, die Ethernet Global Data (S. 2-45) unterstützt.
- Solange der Schlüsselschalterschutz aktiv ist, können Sie mit dem Hand-Programmiergerät nicht die Zeituhr verändern (S. 2-15).
- In Kapitel 2 wurden die Werte hinzugefügt, mit denen die DSM-Konfiguration den Zyklus beeinflusst.
- Für die CPU352 wurden die Unterschiede im Ablauf der Logarithmus-/Exponentialfunktionen (Seite 6-12) und Abschneidefunktionen (S. 11-11) aufgezeigt.
- Die Beschreibung der Ereignisaufzeichnung (SER) wurde überarbeitet, um die Leistungsmerkmale dieser Funktion zu verdeutlichen. Ausführliche Informationen zum Zeitablauf dieser Funktion finden Sie in Anhang A. Diese Funktion steht bei den Ausgabeständen ab 9.0 der CPUs der Serien 35x und 36x zur Verfügung (S. 12-8).
- Für die Ausgabestände ab 9.0 der CPUs der Serien 35x und 36x steht eine neue Bedienanforderung, "Nächsten Ausgabe- und Eingabezyklus überspringen" (SVCREQ #45) zur Verfügung (S. 12-64).
- Die Parameterblocklisten für die Schreib- und Lesefunktionen der Serviceanforderung "Schneller Rückwandplatinen-Statuszugriff" (SVCREQ #46) wurden korrigiert (S. 12-65).
- Weitere Korrekturen und Klarstellungen erfolgten nach Bedarf.

Inhalt dieses Handbuchs

Kapitel 1. Einleitung: Übersicht über die SPS-Systeme Series 90-30, Series 90-20 und Series 90 Micro sowie den Befehlssatz der Series 90-30/20/Micro.

Kapitel 2. Systembetrieb: Beschreibung bestimmter Systemoperationen der SPS-Systeme Series 90-30, Series 90-20 und Series 90 Micro. Hierzu gehören Beschreibungen von SPS-Zyklusablauf, Ein- und Abschaltsequenzen des Systems, Uhren und Zeitgeber, Sicherheit, E/A und Fehlerbehandlung. Ebenfalls enthalten sind allgemeine Informationen, die zu einem grundlegenden Verständnis von Kontaktplanprogrammen führen sollen.

Kapitel 3. Fehlerbeschreibungen und Fehlerbehebung: Fehlersuchinformationen für die SPS-Systeme Series 90-30, 90-20 oder Micro. Hier werden die Fehlerbeschreibung in der SPS-Fehlertabelle sowie die Fehlerkategorien in der E/A-Fehlertabelle beschrieben.

Kapitel 4—12. Befehlssatz der Series 90-30/20/Micro: Beschreibung der für die SPS Series 90-30, Series 90-20 und Series 90 Micro verfügbaren Programmieranweisungen. Diese Kapitel entsprechen den Hauptprogramm-Funktionsgruppen.

Anhang A. Befehlsausführungszeiten: Hier werden für jede Programmieranweisung die Speichergröße in Byte und die Ausführungszeit in Mikrosekunden angegeben. Die Speichergröße gibt an, wieviel Byte von der Funktion in einem Kontaktplan-Anwenderprogramm belegt werden.

Anhang B. Fehlertabellen interpretieren: Dieses Kapitel beschreibt, wie die Meldungsstruktur beim Auslesen der Fehlertabellen mit der Logicmaster 90-30/20/Micro Software interpretiert werden muss.

Anhang C. Mnemonik der Anweisungen: Hier werden die mnemotechnischen Begriffe aufgeführt, die beim Durchsuchen oder Bearbeiten eines Programms zur Anzeige von Programmieranweisungen eingegeben werden können.

Anhang D. Tastenfunktionen: Liste der speziellen Tastaturbelegungen für die Logicmaster 90-30/20/Micro Software. Im Anschluss an diesen Anhang finden Sie eine ***herausnehmbare Übersichtskarte*** dieser Tastenfunktionen und mnemonischen Anweisungen.

Anhang E. Verwendung von Fliesskommazahlen: Hier werden spezielle Betrachtungen zur Verwendung mathematischer Fliesskommaoperationen beschrieben.

Zugehörige Veröffentlichungen

Logicmaster™ 90 Series 90™-30/20/Micro Programmiersoftware, Anwenderhandbuch (GFK-0466).

Logicmaster™ 90 Series 90-30 und 90-20, wichtige Produktinformation (GFK-0468).

SPS Series 90™-30, Installationshandbuch (GFK-0356).

SPS Series 90™-20, Installationshandbuch (GFK-0551).

Series 90™-30 Technische Daten E/A-Module, Handbuch (GFK-0898).

Series 90™ Programmierbares Coprozessormodul und Unterstützungssoftware, Anwenderhandbuch (GFK-0255).

Series 90™ PCM Entwicklungssoftware (PCOP), Anwenderhandbuch (GFK-0487).

CIMPLICITY™ 90-ADS alphanumerisches Anzeigesystem, Anwenderhandbuch (GFK-0499).

CIMPLICITY™ 90-ADS alphanumerisches Anzeigesystem, Referenzhandbuch (GFK-0641).

Alphanumerisches Anzeige-Coprozessormodul, Datenblatt (GFK-0521).

SPS Series 90™-30 und 90-20 Hand-Programmiergerät, Anwenderhandbuch (GFK-0402).

Power Mate APM für SPS Series 90™-30 —Standardmodus, Anwenderhandbuch (GFK-0840).

Power Mate APM für SPS Series 90™-30 —Nachlaufmodus, Anwenderhandbuch (GFK-0781).

Motion Mate™ DSM302 für SPS Series 90™-30, Anwenderhandbuch (GFK-1464)

Series 90™-30 schneller Zähler, Anwenderhandbuch (GFK-0293).

Series 90™-30 Genius-Kommunikationsmodul, Anwenderhandbuch (GFK-0412).

Genius-Kommunikationsmodul, Datenblatt (GFK-0272).

Series 90™-30 Genius™-Buscontroller, Anwenderhandbuch (GFK-1034).

Series 90™-70 FIP-Buscontroller, Anwenderhandbuch (GFK-1038).

Series 90™-30 FIP dezentrale E/A-Zyklussteuerung, Anwenderhandbuch (GFK-1037).

Field Control™ Dezentrales E/A- und Steuerungssystem, Genius™-Busschnittstelle, Anwenderhandbuch (GFK-0825).

SPS Series 90™ Micro, Anwenderhandbuch (GFK-1065).

SPS Series 90™ serielle Kommunikation, Anwenderhandbuch (GFK-0582).

Wir freuen uns über Ihre Kommentare und Vorschläge

Wir bemühen uns bei GE Fanuc Automation, hochwertige technische Dokumentation zu erzeugen. Nehmen Sie sich nach der Benutzung dieses Handbuchs bitte etwas Zeit, die Leserkommentarkarte auf der nächsten Seite auszufüllen und an uns zurückzuschicken.

Libby Allen
Technische Autorin

Abbildungen

A

B

G

H

Kapitel 1 Einleitung

Die speicherprogrammierbaren Steuerungen der Series 90-30, 90-20 und Micro gehören zur Produktfamilie der SPS Series 90 von GE Fanuc. Sie lassen sich einfach installieren und konfigurieren, bieten erweiterte Programmierfunktionen und sind kompatibel zur SPS Series 90-70.

Die SPS Series 90-20 bietet eine kostengünstige Plattform für Anwendungen mit einer geringen Anzahl von E/A-Punkten. Sie ist hauptsächlich auf folgende Punkte ausgerichtet:

- Eine kleine SPS, die man einfach benutzen, installieren, erweitern und warten kann.
- Eine kostengünstige, familienkompatible SPS.
- Einfachere Systemintegration durch Standard-Kommunikationshardware und -Protokolle.

Die SPS Series 90 Micro bietet ebenfalls eine kostengünstige Plattform für Anwendungen mit einer geringeren Anzahl von E/A-Punkten. Die Hauptanwendungsziele der Micro SPS sind die gleichen wie die der Series 90-20. Darüber hinaus bietet die Micro noch folgende Funktionen:

- Bei der Micro SPS sind CPU, Stromversorgung und Ein- und Ausgänge in einem einzigen kleinen Gerät integriert.
- Die meisten Modelle besitzen einen schnellen Zähler.
- Durch die Zusammenfassung von CPU, Stromversorgung und Ein- und Ausgängen in einem Gerät ist die Micro sehr einfach zu konfigurieren.

Die Softwarestrukturen für die SPS-Modelle der Series 90-30 bis einschließlich CPU 341 und die Series 90-20 verwenden eine Architektur, die Speicher- und Ausführungsprioritäten im 80188 Mikroprozessor verwaltet. Die Modelle 35x und 36x benutzen hierfür einen 80386 EX Mikroprozessor, die SPS Series 90 Micro den H8 Mikroprozessor. Diese Operation unterstützt sowohl die Programmausführung als auch die grundlegenden Organisationsarbeiten, wie zum Beispiel Diagnoseroutinen, E/A-Aktualisierung und Alarmbearbeitung. Die Systemsoftware enthält auch Routinen zum Datenaustausch mit dem Programmiergerät, die für Speichern und Laden der Anwenderprogramme, Rückmeldung von Statusinformationen und Steuerung der SPS verwendet werden können.

In der SPS Series 90-30 wird das Anwenderprogramm, das den an die SPS angeschlossenen Prozess steuert, durch einen dedizierten Befehlsfolge-Coprozessor (ISCP) gesteuert. Bei den Modellen 313 und höher ist der ISCP hardwaremäßig ausgeführt, bei den Modellen 311 und der Micro SPS wird eine softwaremäßige Ausführung benutzt. Der 80188 Mikroprozessor und der ISCP können simultan arbeiten, wodurch der Mikroprozessor den Datenverkehr abwickeln kann, während der ISCP den Hauptteil des Anwenderprogramms bearbeitet. Der Mikroprozessor muss dabei allerdings die nichtbooleschen Funktionsblöcke bearbeiten.

In den speicherprogrammierbaren Steuerungen Series 90-30/90-20/Micro liegt ein Fehler vor, wenn bestimmte Ausfälle oder Zustände auftreten, die Betrieb und Leistungsfähigkeit des Systems beein-

trächtigen. Diese Zustände können die Fähigkeit des Systems mindern, eine Maschine oder einen Prozess zu steuern. Andere Zustände können nur als Warnung dienen (z.B. die Meldung einer schwachen Batterie, die anzeigt, dass die zur Speicherpufferung erforderliche Spannung zu niedrig ist und die Batterie ausgewechselt werden muss). Der Ausfall oder Zustand wird Fehler genannt.

Fehler werden durch eine Alarmbearbeitungsfunktion in der Software bearbeitet, die den Fehler in die SPS-Fehlertabelle oder in die E/A-Fehlertabelle einträgt. Bei dem Modell 331 und den höheren CPU-Modellen werden die Fehler auch mit einem Zeitstempel versehen. Diese Tabellen können mit den Steuerungs- und Statusfunktionen im SPS-Fehlertabellenmenü bzw. im E/A-Fehlertabellenmenü der Logicmaster 90-30/90-20/Micro Software angezeigt werden.

Hinweis

Fliesskommafunktionen werden ***nur*** bei CPUs der Modelle 35x und 36x, Ausgabestand 9 oder höher und bei allen Ausgabeständen der CPU352 unterstützt.

Die CPU Modell 364 (Ausgabestand 9.10 oder höher) ist die einzige CPU der Series 90-30, die EGD unterstützt.

Hinweis

Weitere Informationen finden Sie hinten in diesem Handbuch.

- In Anhang A werden für jede Programmieranweisung die Speichergröße in Byte und die Ausführungszeit in Mikrosekunden angegeben.
- Anhang B beschreibt die Interpretation des Nachrichtenstrukturformats beim Lesen der SPS- und E/A-Fehlertabellen.
- Anhang C enthält eine Liste der mnemonischen Formen der Anweisungen zum Durchsuchen oder Bearbeiten eines Programms.
- Anhang D enthält eine Liste der in der Logicmaster 90-30/20/Micro Software benutzten speziellen Tastaturbelegungen.
- Anhang E beschreibt die Verwendung arithmetischer Fliesskommaoperationen.

Kapitel 2 Systembetrieb

In diesem Kapitel werden bestimmte Systemoperationen der SPS Series 90-30, 90-20 und Micro beschrieben. Hierzu gehören:

Abschnitt 1: SPS-Zyklus - Zusammenfassung

Das Programm in einer SPS Series 90-30, 90-20 oder Micro läuft solange zyklisch ab, bis es durch einen Befehl vom Programmiergerät oder von einem anderen Gerät gestoppt wird. Die für die einmalige Ausführung eines Programms erforderliche Abfolge von Operationen wird Zyklus genannt. Zusätzlich zur Bearbeitung des Programms werden in einem Zyklus noch die Daten von den Eingabegeräten erfasst, Daten an Ausgabegeräte geschickt, interne Organisationsaufgaben erledigt, das Programmiergerät bedient und andere Kommunikationsaufgaben erledigt.

Normalerweise arbeiten die speicherprogrammierbaren Steuerungen Series 90-30, 90-20 und Micro im Modus **STANDARD-PROGRAMMZYKLUS**. Weitere Betriebsarten sind **STOP MIT GESPERRTER E/A** , **STOP MIT FREIGEGEBENER E/A** und **KONSTANTER ZYKLUS**. Jede dieser Betriebsarten (die in diesem Kapitel beschrieben werden) wird durch externe Ereignisse und Einstellungen der Anwendungskonfiguration gesteuert. Die SPS wählt die Betriebsart zu Beginn jedes Zyklus aus.

Standard-Programmzyklus

STANDARD-PROGRAMMZYKLUS läuft normalerweise unter allen Bedingungen. Die CPU bearbeitet das Anwenderprogramm, aktualisiert die E/A und führt Kommunikation und andere Aufgaben durch. Dies läuft repetitiv im CPU-Zyklus ab. Die Bearbeitungsfolge des Standard-Programmzyklus besteht aus sieben Teilen:

1. Organisation Zyklusstart
2. Eingabezyklus (Eingänge lesen)
3. Anwenderprogramm bearbeiten
4. Ausgabezyklus (Ausgänge aktualisieren)
5. Programmiergerät bedienen
6. Andere Geräte bedienen
7. Diagnoseroutinen

All diese Schritte werden in jedem Zyklus durchlaufen. Obwohl sich das Programmiergeräte-Kommunikationsfenster in jedem Zyklus öffnet, wird das Programmiergerät nur dann bedient, wenn ein Modulfehler erkannt wurde oder das Programmiergerät eine Bedienanforderung abschickt. Das bedeutet, dass das Programmiergeräte-Kommunikationsfenster zuerst prüft, ob es etwas zu tun gibt. Liegen keine Aufgaben vor, schließt es sich wieder. Die nachstehende Abbildung zeigt den Ablauf des Standard-Programmzyklus.

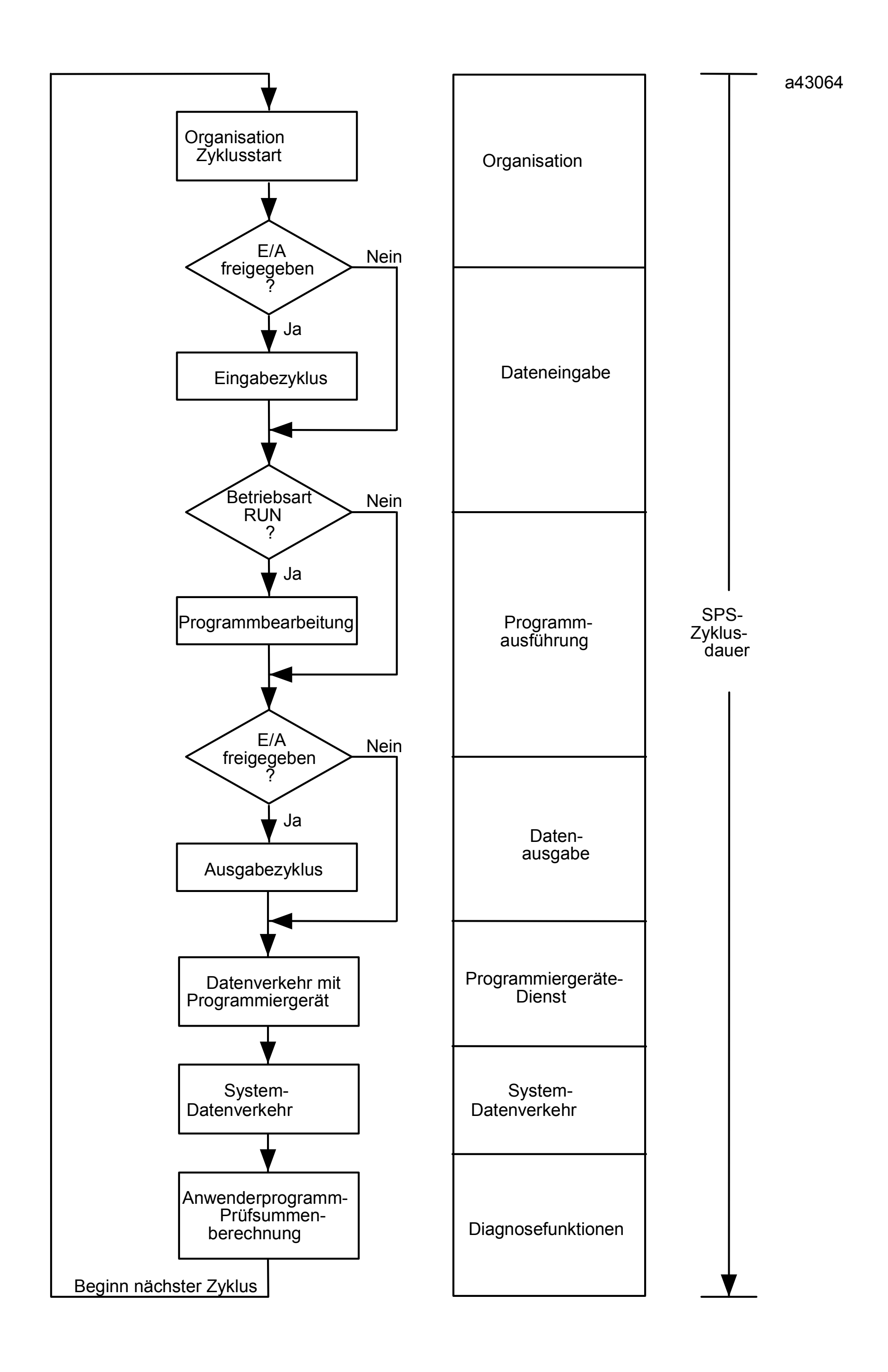


Abbildung 2-1. SPS-Zyklus

Wie Sie in der SPS-Zyklusfolge sehen können, enthält der Zyklus mehrere Elemente. Diese Elemente tragen entsprechend der nachstehenden Tabelle zur Gesamt-Zyklusdauer bei.

Tabelle 2-1. Beiträge zur Zyklusdauer

Zykluselement	Beschreibung		Bearbeitungszeit (ms)[4]
			Modelle 351, 352, 350 und 36x (die Zeiten für die Modelle 350 und 36x werden als gleich angenommen)
Organisation	• Zyklusdauer berechnen. • Start des nächsten Zyklus planen. • Modus für nächsten Zyklus festlegen. • Fehlerreferenztabelle aktualisieren. • Zeitüberwachung rücksetzen.		0,279
Dateneingabe	Eingangsdaten von Eingangs- und Zusatzmodulen werden empfangen.		Siehe Tabelle 2-2 für Beitrag zur Zykluszeit.
Programm-bearbeitung	Anwenderprogramm wird bearbeitet.		Die Bearbeitungszeit ist abhängig von der Programmlänge und den im Programm verwendeten Befehlstypen. Die Befehlsausführungszeiten finden Sie in Anhang A.
Datenausgabe	Ausgangsdaten werden zu Ausgangs- und Zusatzmodulen geschickt.		Siehe Tabelle 2-2 für Beitrag zur Zykluszeit.
Bedienen externer Geräte	Bedienanforderungen von Programmiergeräten und intelligenten Modulen werden bearbeitet.[1]	HHP	0,334
		LM-90	0,517
		PCM[2]	0,482
Neukonfiguration	Steckplätze mit fehlerhaften Modulen und leere Steckplätze werden überwacht.		0,319[6]
Diagnosefunktionen	Überprüfung der Anwenderprogramm-Integrität		Pro Zyklus 0,010 pro Wort, über das die Prüfsumme gebildet wird[3, 7]

1. Der Beitrag externer Geräte zur Zyklusdauer hängt von der Betriebsart des Kommunikationsfensters ab, in dem der Dienst bearbeitet wird. Bei Fenstermodus **LIMITED** [einschränkt] werden für diese Fenster bei den CPUs 311, 313, 323 und 331 maximal 8 Millisekunden aufgewandt. Bei den CPUs 340 und höher sind es 6 Millisekunden. Bei Fenstermodus **RUN-TO-COMPLETION** [vollständige Bearbeitung] können je nach Anzahl gleichzeitig gestellter Anforderungen bis zu 50 ms für dieses Fenster aufgewandt werden.
2. Diese Messungen wurden durchgeführt mit einem physikalisch vorhandenen PCM, das aber nicht konfiguriert war und in dem keine Anwendertasks abliefen.
3. Die Anzahl der Worte, die in einem Zyklus in die Prüfsumme eingeschlossen werden, kann mit dem Funktionsblock SVCREQ verändert werden.
4. Diese Messungen wurden mit einem leeren Programm und der Standardkonfiguration durchgeführt. Die SPS der Series 90-30 waren in einem leeren Chassis ohne angeschlossene Erweiterungschassis. Bei den in dieser Tabelle angegebenen Werten wird auch angenommen, dass kein periodisch ablaufendes Unterprogramm aktiv ist. Die Zeiten werden länger, wenn ein solches Unterprogramm aktiv ist.
5. Die Dateneingabezeit für die Micro SPS kann wie folgt bestimmt werden: 0,365 ms (fester Zyklus) + 0,036 ms (Filterzeit) x (Gesamtzykluszeit)/0,5 ms.
6. Da die Micro SPS einen festen Satz E/A besitzt, ist eine Neukonfiguration nicht erforderlich.
7. Das Anwenderprogramm der Micro SPS liegt im Flash Memory. Es wird daher nicht auf seine Integrität untersucht.

Tabelle 2-2. Beitrag zur E/A-Zykluszeit bei den Modellen 90-30 35x und 36x (in Millisekunden)

		CPU-Modelle 35x und 36x		
Modultyp		Haupt-chassis	Erweite-rungs-chassis	Dezentrales Chassis
Diskrete Eingänge, 8 Punkte		0,030	0,055	0,206
Diskrete Eingänge, 16 Punkte		0,030	0,055	0,206
Diskrete Eingänge, 32 Punkte		0,043	0,073	0,269
Diskrete Ausgänge, 8 Punkte		0,030	0,053	0,197
Diskrete Ausgänge, 16 Punkte		0,030	0,053	0,197
Diskrete Ausgänge, 32 Punkte		0,042	0,070	0,259
Kombination diskrete Eingänge/Ausgänge		0,060	0,112	0,405
Analogeingänge, 4 Kanäle		0,075	0,105	0,396
Analogausgänge, 2 Kanäle		0,058	0,114	0,402
Analogeingänge, 16 Kanäle (Strom oder Spannung)		0,978	1,446	3,999
Analogausgänge, 8 Kanäle		1,274	1,988	4,472
Kombination analoge Eingänge/Ausgänge		1,220	1,999	4,338
Schneller Zähler		1,381	2,106	5,221
E/A-Prozessormodul		1,574	2,402	6,388
Ethernet-Schnittstelle (ohne Anschluss)		0,038	0,041	0,053
Power Mate APM (1 Achse)		1,527	2,581	6,388
Power Mate APM (2 Achsen)		1,807	2,864	7,805
DSM 302 *	40 AI, 6 AQ	2,143	3,315	9,527
	50 AI, 9 AQ	2,427	3,732	11,092
	64 AI, 12 AQ	2,864	4,317	13,138
GCM	ohne Geräte	0,911	1,637	5,020
	8 64-Wort-Geräte	8,826	16,932	21,179
GCM+	ohne Geräte	0,567	0,866	1,830
	32 64-Wort-Geräte	1,714	2,514	5,783
GBC	ohne Geräte	0,798	1,202	2,540
	32 64-Wort-Geräte	18,382	25,377	70,777
PCM 311	nicht konfiguriert oder keine Anwendung	0,476	--	--
	128 %R schnellstmög-lich lesen	0,485	--	--
ADC (ohne Task)		0,476	--	--
Schnittstellenmo-dul, Master	ohne Geräte	0,569	0,865	1,932
	16 64-Punkt-Geräte	4,948	7,003	19,908
Schnittstellenmo-dul, Slave	32 Punkte	0,087	0,146	0,553
	64 Punkte	0,154	0,213	0,789

* Bei Anwendungen, bei denen die Beiträge des DSM zur Bearbeitungszeit den Maschinenbetrieb beeinträchtigen, müssen Sie ggf. den Funktionsblock "E/A-Aktualisierung" sowie die Dienstanforderungen "Unterdrückung der E/A-Aktualisierung" und "Schneller Rückwandplatinen-Statuszugriff" einsetzen, um die erforderlichen Daten zu und vom Bewegungsmodul zu übertragen, ohne bei jedem Zyklus alle Daten zu erhalten. Weitere Angaben finden Sie unter *Motion Mate DSM302 für SPS Series 90-30, Anwenderhandbuch*, GFK-1464.

Tabelle 2-3. Beitrag zur E/A-Zykluszeit bei der Series 90-30 bis zu Modell 341x (in Millisekunden)

Modultyp		CPU-Modell						
			331			340/341		
		311/313	Haupt-chassis	Erweite-rungschassis	Dezentrales Chassis	Haupt-chassis	Erweite-rungschassis	Dezentrales Chassis
Diskrete Eingänge, 8 Punkte		0,076	0,054	0,095	0,255	0,048	0,089	0,249
Diskrete Eingänge, 16 Punkte		0,075	0,055	0,097	0,257	0,048	0,091	0,250
Diskrete Eingänge, 32 Punkte		0,094	0,094	0,126	0,335	0,073	0,115	0,321
Diskrete Ausgänge, 8 Punkte		0,084	0,059	0,097	0,252	0,053	0,090	0,246
Diskrete Ausgänge, 16 Punkte		0,083	0,061	0,097	0,253	0,054	0,090	0,248
Diskrete Ausgänge, 32 Punkte		0,109	0,075	0,129	0,333	0,079	0,114	0,320
Kombination Ein-/Ausgänge, 8 Pkt.		0,165	0,141	0,218	0,529	0,098	0,176	0,489
Analogeingänge, 4 Kanäle		0,151	0,132	0,183	0,490	0,117	0,160	0,462
Analogausgänge, 2 Kanäle		0,161	0,138	0,182	0,428	0,099	0,148	0,392
Schneller Zähler		2,070	2,190	2,868	5,587	1,580	2,175	4,897
Power Mate APM (1 Achse)		2,330	2,460	3,175	6,647	1,750	2,506	5,899
Power Mate APM (2 Achsen)		3,181	3,647	4,497	9,303	2,154	3,097	7,729
DSM 302	40 AI, 6 AQ	3,613	4,081	5,239	11,430	2,552	3,648	9,697
	50 AI, 9 AQ	4,127	4,611	5,899	13,310	2,911	4,170	11,406
	64 AI, 12 AQ	4,715	5,276	6,759	15,747	3,354	4,840	13,615
GCM	ohne Geräte	0,041	0,054	0,063	0,128	0,038	0,048	0,085
	8 64-Punkt-Geräte	11,420	11,570	13,247	21,288	9,536	10,648	19,485
GCM+	ohne Geräte	0,887	0,967	1,164	1,920	0,666	0,901	1,626
	32 64-Punkt-Geräte	4,120	6,250	8,529	21,352	5,043	7,146	20,052
PCM 311	nicht konfiguriert oder keine Anwendung	--	3,350	--	--	1,684	--	--
	128 %R schnellstmöglich lesen	--	4,900	--	--	2,052	--	--
ADC 311		--	3,340	--	--	1,678	--	--
Analogeingänge, 16 Kanäle (Strom oder Spannung)		1,370	1,450	1,937	4,186	1,092	1,570	3,796
Schnittstellen-modul, Master	ohne Geräte	1,910	2,030	1,169	1,925	0,678	0,904	1,628
	16 64-Punkt-Geräte	6,020	6,170	8,399	21,291	4,992	6,985	20,010
Schnittstellen-modul, Slave	32 Punkte	0,206	0,222	0,289	0,689	0,146	0,226	0,636
	64 Punkte	0,331	0,350	0,409	1,009	0,244	0,321	0,926

* Bei Anwendungen, bei denen die Beiträge des DSM zur Bearbeitungszeit den Maschinenbetrieb beeinträchtigen, müssen Sie ggf. den Funktionsblock "E/A-Aktualisierung" sowie die Dienstanforderungen "Unterdrückung der E/A-Aktualisierung" und "Schneller Rückwandplatinen-Statuszugriff" einsetzen, um die erforderlichen Daten zu und vom Bewegungsmodul zu übertragen, ohne bei jedem Zyklus alle Daten zu erhalten. Weitere Angaben finden Sie unter *Motion Mate DSM302 für SPS Series 90-30, Anwenderhandbuch*, GFK1464.

Berechnung der Zykluszeit

In Tabelle 2-1 sind die sieben Elemente dargestellt, die zur SPS-Zykluszeit beitragen. Die Zykluszeit setzt sich zusammen aus festen (Organisation und Diagnosefunktionen) und aus variablen Zeiten. Die variablen Zeiten ändern sich je nach E/A-Konfiguration, Programmgröße und dem Typ des an der SPS angeschlossenen Programmiergerätes.

Beispiel einer Zykluszeitberechnung

Die nachstehende Tabelle zeigt an einem Beispiel, wie Sie die Zykluszeit einer SPS Series 90-30 Modell 331 berechnen können.

Für diese Berechnung wurden folgende Module und Anweisungen zugrunde gelegt:

- Eingangsmodule: Fünf Series 90-30 Eingangsmodule mit je 16 Punkten.
- Ausgangsmodule: Vier Series 90-30 Ausgangsmodule mit je 16 Punkten.
- Programmieranweisungen: Ein Programm aus 1200 Schritten mit 700 Booleschen Anweisungen (LD, AND, OR, etc.), 300 Ausgangsmerkern (OUT, OUTM, usw.), und 200 arithmetischen Funktionen (ADD, SUB, usw.).

Organisation

Im Organisationsteil des Zyklus werden alle Tasks ausgeführt, die für die Vorbereitung des Zyklusstarts benötigt werden. Befindet sich die SPS in der Betriebsart **KONSTANTER ZYKLUS** , wird der Zyklus solange verzögert, bis die eingestellte Zykluszeit verstrichen ist. Ist diese Zykluszeit bereits abgelaufen, wird der Kontakt OV_SWP %SA0002 gesetzt und der Zyklus wird unverzüglich fortgesetzt. Als nächstes werden die Zeitgeberwerte (1/100, 1/10 und 1 s) aktualisiert, indem der Zeitunterschied zum vorherigen Zyklus und die neue Zykluszeit berechnet werden. Damit keine Genauigkeit verloren geht, wird der tatsächliche Zyklusbeginn in Schritten von 100 Mikrosekunden aufgezeichnet. Jeder Zeitgeber besitzt ein Restfeld mit der Anzahl von 100-Mikrosekunden-Schritten, die seit der letzten Erhöhung des Zeitgeberwerts aufgelaufen sind.

Eingabezyklus

Die Eingänge werden im Eingabeteil des Zyklus abgefragt, unmittelbar vor der Programmbearbeitung. Während dieses Zyklusteils werden alle Series 90-30 Eingangsmodule abgefragt und ihre Daten im %I- (diskrete Eingänge) bzw. %AI- (analoge Eingänge) Speicher abgelegt. Von einem Genius-Kommunikationsmodul, einem erweiterten Genius-Kommunikationsmodul oder einem Genius-Buscontroller empfangene Global Data werden im %G-Speicher abgelegt.

Die Module werden in steigender Reihenfolge der Referenzadressen abgefragt, beginnend mit dem Genius-Kommunikationsmodul, dann den diskreten Eingangsmodulen und zuletzt den analogen Eingangsmodulen.

Ist die CPU in **STOP**-Modus und wurde sie so konfiguriert, dass sie im **STOP**-Modus die Eingänge nicht abfragt, wird der Eingabezyklus übersprungen.

Anwenderprogramm-Bearbeitungszyklus

Im Anwenderprogrammzyklus findet die eigentliche Programmausführung statt. Die Programmbearbeitung beginnt immer mit der ersten Anweisung im Anwenderprogramm, unmittelbar nach Beendigung des Eingabezyklus Die Bearbeitung der Logik erzeugt einen neuen Satz Ausgangsdaten. Die Programmbearbeitung endet, wenn die Anweisung END ausgeführt wird (END ist unsichtbar, wenn Sie keinen Handmonitor benutzen).

Das Anwenderprogramm wird von ISCP und 80C188 Mikroprozessor bearbeitet. Bei den CPUs von Modell 313 aufwärts führt der ISCP die Booleschen Anweisungen aus, während der 80C188 oder 80386EX Zeitgeber, Zähler und Funktionsblöcke bearbeitet. Bei den CPUs der Modelle 311 und bei der Series 90-20 bearbeitet der 80C188 alle Booleschen Anweisungen, Zeitgeber, Zähler und Funktionsblöcke. In der Micro bearbeitet der H8-Prozessor alle Booleschen Anweisungen und Funktionsblöcke.

Anhang A enthält eine Liste der Ausführungszeiten der einzelnen Programmierfunktionen.

Ausgabezyklus

Die Aktualisierung der Ausgänge erfolgt unmittelbar nach der Programmbearbeitung im Ausgabezyklus. Hierzu werden Daten aus dem %Q- (für digitale Ausgänge) bzw. dem %AQ- (für analoge Ausgänge) Speicher verwendet. Wurde das Genius-Kommunikationsmodul für das Aussenden von Global Data konfiguriert, werden Daten aus dem %G-Speicher zum GCM, GCM+ oder GBC geschickt. Bei Series 90-20 und Micro werden nur digitale Ausgänge aktualisiert.

Im Ausgabezyklus werden alle Series 90-30 Ausgangsmodule in steigender Adressreihenfolge aktualisiert.

Ist die CPU im **STOP**-Modus und wurde sie so konfiguriert, dass sie im **STOP**-Modus die Ausgänge nicht aktualisiert, wird der Ausgabezyklus übersprungen. Der Ausgabezyklus ist beendet, wenn alle Ausgabedaten zu allen Series 90-30 Ausgangsmodulen geschickt wurden.

Berechnung der Programm-Prüfsumme

Am Ende jedes Zyklus wird für das Anwenderprogramm eine Prüfsummenberechnung durchgeführt. Da eine Berechnung der Prüfsumme für das ganze Programm zu viel Zeit in Anspruch nehmen würde, können Sie im CPU-Detailmenü die Anzahl der Worte (8 bis 32) festlegen, die in die Prüfsumme einbezogen werden.

Stimmt die berechnete Prüfsumme nicht mit der Referenz-Prüfsumme überein, wird ein Programmprüfsummen-Ausnahmemerker gesetzt. Hierdurch erfolgt ein Fehlereintrag in die SPS-Fehlertabelle und der SPS-Modus wechselt auf **STOP**. Das Programmiergeräte-Kommunikationsfenster ist von einem Prüfsummenfehler nicht betroffen. In der Grundeinstellung werden 8 Worte in die Prüfsumme einbezogen.

Programmiergeräte-Kommunikationsfenster

Dieser Teil des Zyklus ist für den Datenaustausch mit dem Programmiergerät bestimmt. Ist ein Programmiergerät angeschlossen, dann führt die CPU das Programmiergeräte-Kommunikationsfenster aus. Dieses Fenster bleibt geschlossen, wenn kein Programmiergerät angeschossen ist und kein Modul konfiguriert werden muss. Pro Zyklus wird jeweils nur ein Modul konfiguriert.

Unterstützung wird gegeben für das Hand-Programmiergerät und für andere Programmiergeräte, die über das Series Ninety Protocol (SNP) an den seriellen Port angeschlossen werden können. Ebenfalls unterstützt wird der Datenaustausch zwischen Programmiergerät und intelligenten Zusatzmodulen.

Im standardmäßig eingestellten begrenzten Fenstermodus führt die CPU bei jedem Zyklus eine Operation für das Programmiergerät durch. Das heißt, sie antwortet auf eine Bedienanforderung oder auf einen Tastendruck. Stellt das Programmiergerät eine Anforderung, deren Bearbeitung mehr als 6 ms (oder, je nach CPU, 8 ms - siehe Hinweis) in Anspruch nimmt, wird die Bearbeitung dieser Anforderung so über mehrere Zyklen verteilt, dass jeder Zyklus nur mit maximal 6 ms (oder, je nach CPU, 8 ms - siehe Hinweis) betroffen ist.

Hinweis

Die Maximaldauer für das Kommunikationsfenster beträgt 6 ms für CPUs ab 340 aufwärts, bzw. 8 ms für die Modelle 311, 313, 323 und 331.

Die nachstehende Abbildung zeigt, wie der Programmiergeräte-Kommunikationsteil des Zyklus abläuft.

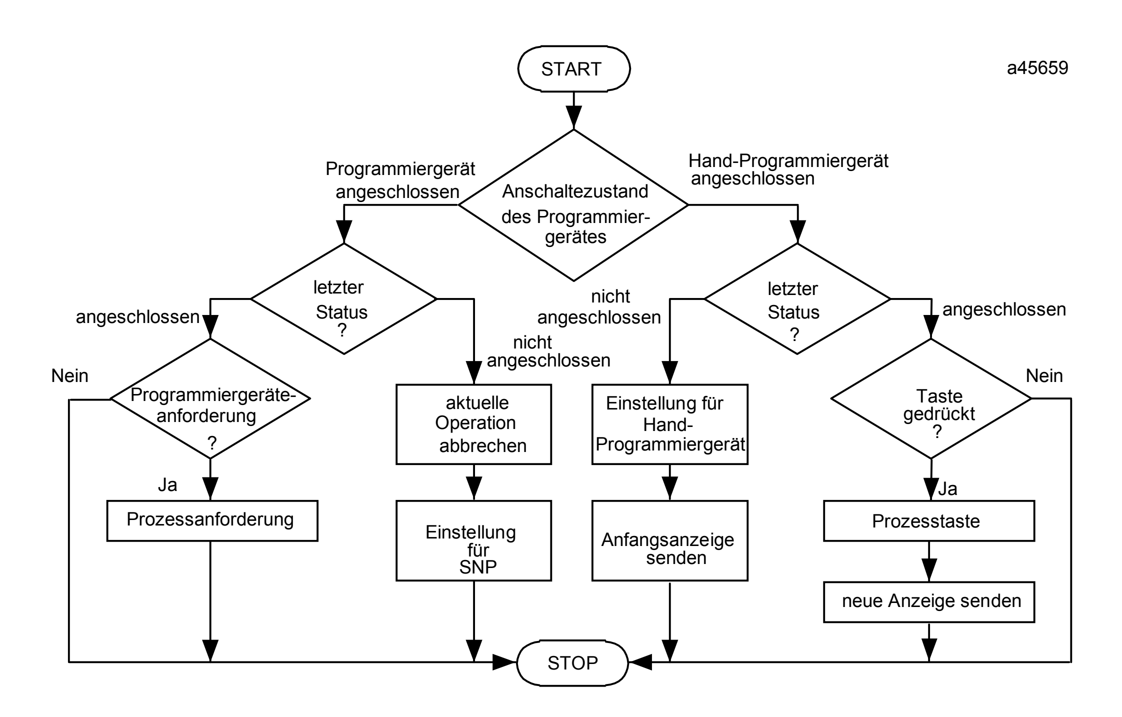


Abbildung 2-2. Programmiergeräte-Kommunikationsfenster - Ablaufdiagramm

System-Kommunikationsfenster

In diesem Zyklusabschnitt werden die Kommunikationsanforderungen von intelligenten Zusatzmodulen (z.B. PCM oder DSM) bearbeitet (siehe Ablaufdiagramm). Die Anforderungen werden in der Reihenfolge ihres Auftretens bearbeitet. Da intelligente Zusatzmodule aber mit umlaufender Berechtigung abgefragt werden, besitzt kein intelligentes Zusatzmodul eine Priorität über ein anderes.

Im standardmäßig eingestellten Modus `vollständige Bearbeitung` ist die Länge des System-Kommunikationsfensters auf 50 Millisekunden beschränkt. Stellt ein intelligentes Zusatzmodul eine Anforderung, deren Bearbeitung mehr als 50 ms in Anspruch nimmt, wird die Anforderung so über mehrere Zyklen verteilt, dass kein Zyklus mit mehr als 50 ms daran beteiligt ist.

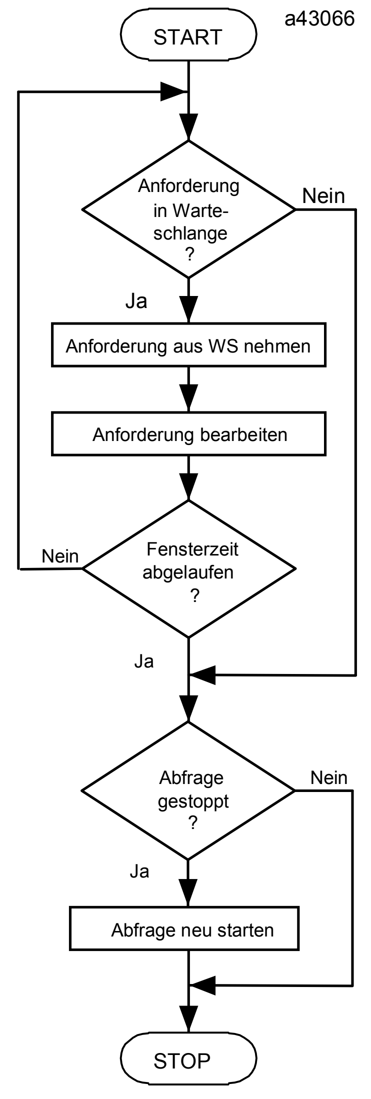


Abbildung 2-3. System-Kommunikationsfenster - Ablaufdiagramm

PCM-Datenverkehr mit der SPS (Modelle 331 und höher)

Intelligente Zusatzmodule (z.B. das PCM) können die CPU nicht unterbrechen, wenn sie bedient werden wollen. Die CPU muss die einzelnen intelligenten Zusatzmodule zyklisch nach Bedienanforderungen abfragen. Diese Abfrage findet während des Zyklus asynchron im Hintergrund statt (siehe nachstehendes Ablaufdiagramm).

Wird ein intelligentes Zusatzmodul abgefragt und sendet der CPU eine Bedienanforderung, wird die Anforderung zur Bearbeitung im System-Kommunikationsfenster in eine Warteschlange eingereiht.

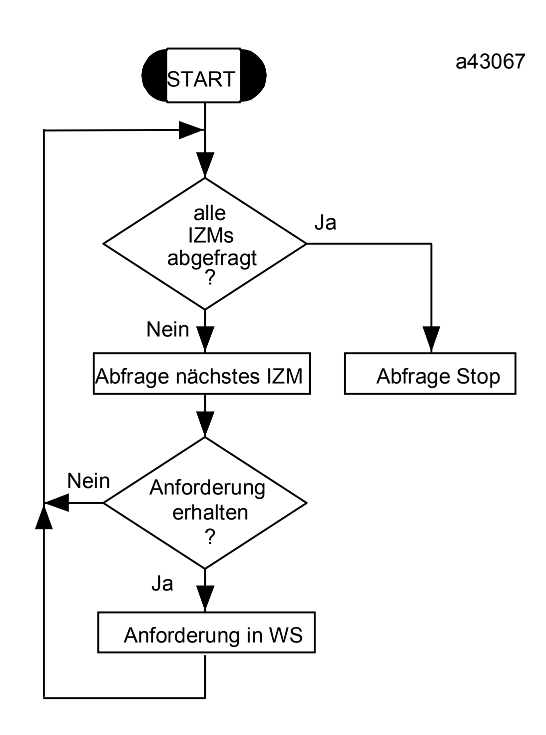


Abbildung 2-4. PCM-Kommunikation mit der SPS

DSM-Datenverkehr mit der

Das DSM302 ist ein intelligentes Modul, das asynchron zum CPU-Modul der Series 90-30 arbeitet. Die Daten werden automatisch zwischen CPU und DSM ausgetauscht.

Das DSM kann für drei unterschiedliche Längen von %AI- und %AQ-Daten konfiguriert werden. Eine SPS CPU benötigt Zeit, um Daten über die Rückwandplatine mit dem DSM302 auszutauschen. In Tabelle 2-2 finden Sie für die unterschiedlichen Konfigurationen von %AI- und %AQ-Daten die zeitliche Auswirkung auf den Zyklus. Weitere zeitliche Betrachtungen zum DSM302-Modul finden Sie in *Motion Mate DSM302 für SPS Series 90-30, Anwenderhandbuch*, GFK-1464.

Varianten des Standard-Programmzyklus

Über die normale Ausführung des Standard-Programmzyklus hinaus können bestimmte Varianten auftreten oder erzwungen werden. Diese Varianten, die in den nächsten Absätzen beschrieben werden, können aus der Programmiersoftware heraus angezeigt und verändert werden.

Betriebsart "konstante Zyklusdauer"

Im Standard-Programmzyklus wird jeder Zyklus so schnell wie möglich abgearbeitet, wobei die einzelnen Zyklen unterschiedlich lang sind. Eine Alternative hierzu ist **KONSTANTE ZYKLUSDAUER**, bei der jeder Zyklus gleich lang ist. Sie müssen hierzu die konfigurierte konstante Zykluszeit einstellen, die dann Standard-Zyklusbetriebsart wird und bei jedem Wechsel der SPS von **STOP** auf **RUN** zum Tragen kommt. Sie können den Zeitgeber für konstante Zykluszeit auf einen Wert zwischen 5 und 200 ms (bis zu 500 ms bei den CPUs 35x und 36x) einstellen. Die Standardeinstellung ist 100 ms.

Wegen Schwankungen in der für die verschiedenen Teile des SPS-Zyklus benötigten Zeit sollten Sie die konstante Zykluszeit auf einen Wert einstellen, der mindestens 10 Millisekunden höher ist als der auf der Statuszeile für die Betriebsart **NORMALZYKLUS** angezeigte Wert. Hierdurch verhindern Sie, dass Zyklusüberschreitungsfehler im Übermaß auftreten.

Die Betriebsart KONSTANTE ZYKLUSDAUER kann zum Beispiel verwendet werden, wenn E/A-Punkte oder Register in konstanten Zeitabständen abgefragt werden müssen (z.B. in Regelungsalgorithmen). Bei einem Anwendungsfall der Betriebsart **KONSTANTE ZYKLUSDAUER** kann zum Beispiel sichergestellt werden, dass die E/A in festen Zeitabständen aktualisiert werden. Eine andere Anwendung ist, wenn zwischen Ausgabezyklus und nächstem Eingabezyklus eine bestimmte Zeitspanne verstreichen muss, damit die Eingänge nach dem Empfang von Ausgabedaten aus dem Programm einschwingen können.

Läuft der Zeitgeber für konstante Zyklusdauer ab, ehe der Zyklus abgeschlossen ist, wird der gesamte Zyklus einschließlich der Fenster beendet. Zu Beginn des nächsten Zyklus wird jedoch ein Zyklusüberschreitungsfehler eingetragen.

Hinweis

Im Gegensatz zur aktiven konstanten Zyklusdauer, die nur im **RUN**- Modus editiert werden kann, kann die konfigurierte konstante Zyklusdauer nur im **STOP** -Modus editiert werden. Die gemachten Änderungen werden erst wirksam, nachdem Sie die Funktion "Konfiguration vom Programmiergerät in die SPS speichern” ausgeführt haben. Nach der Speicherung wird dies zur Standard-Zyklusbetriebsart.

SPS-Zyklus in STOP-Modus

Das Anwenderprogramm wird nicht bearbeitet, wenn die SPS im **STOP**-Modus ist. Die Kommunikation mit dem Programmiergerät und intelligenten Zusatzmodulen wird nicht unterbrochen. Ebenfalls wird die Abfrage fehlerhafter Module und die Ausführung der Modulumkonfiguration im **STOP**-Modus fortgesetzt. Die vom Betriebssystem benutzten Zeitscheiben sind größer als im **RUN**-Modus (normalerweise etwa 50 Millisekunden pro Fenster). Sie können wählen, ob die E/A aktuali-

siert werden. Die E/A-Zyklen können im **STOP**-Modus ausgeführt werden, wenn der Parameter ***IOScan-Stop*** im CPU-Detailmenü auf **YES** gesetzt wurde.

Kommunikationsfenster-Betriebsarten

Standardmäßig ist das Programmiergeräte-Kommunikationsfenster im "begrenzten" Modus. Das bedeutet, dass jede Anforderung, deren Bearbeitung länger als 6 ms in Anspruch nimmt, so über mehrere Zyklen hinweg bearbeitet wird, dass kein Zyklus mehr als 6 ms lang in Anspruch genommen wird. Bei den CPUs 313, 323 und 331 kann beim Speichern im **RUN**-Modus der Zyklus 12 ms lang beansprucht werden. Der aktive Fenstermodus kann über das Logicmaster-Menü "Zyklussteuerung" verändert werden — Hinweise zur Änderung des aktiven Fenstermodus finden Sie in Kapitel 5 "SPS-Steuerung und Status" von *Logicmaster 90™ Series 90™-30/20/Micro Programmiersoftware, Anwenderhandbuch* (GFK-0466).

Hinweis

Wird der Systemfenstermodus auf "begrenzt" umgeschaltet, wird zwar die Auswirkung von Zusatzmodulen (z.B. PCM oder GBC), die mit der SPS über das Systemfenster Daten austauschen, auf die Zykluszeit reduziert, die Reaktion auf deren Anforderungen wird aber auch langsamer.

Schlüsselschalter an CPUs 35x und 36x: Betriebsartenwechsel und Schutz von Flash Memory

Über einen Schlüsselschalter auf der Vorderseite der CPUs 35x und 36x können Sie für den Flash Memory Schreibschutz einstellen. Stellen Sie hierzu den Schalter auf **ON**/**RUN**. Der Flash Memory kann nur überschrieben werden, wenn der Schalter wieder auf OFF zurückgestellt wird.

Ab Ausgabestand 7 der CPUs 351 und 352 besitzt dieser Schlüsselschalter noch eine weitere Funktion: Mit ihm können Sie die SPS in **STOP**- oder **RUN** -Modus schalten und nicht fatale Fehler löschen (siehe nächsten Abschnitt).

Ab Ausgabestand 8 der CPUs 351 und 352 besitzt dieser Schlüsselschalter eine erweiterte Speicherschutzfunktion: Sie können nun zwei zusätzliche Arten von Speicherschutz aktivieren (siehe Abschnitt "Speicherschutz ab Ausgabestand 8").

Ist der Schlüsselschalter aktiviert und in Stellung ON/RUN, können Sie die Zeituhr nur über die Programmiersoftware einstellen. Solange der Schlüsselschalterschutz aktiv ist, können Sie die Zeituhr nicht mit dem Hand-Programmiergerät verändern.

Schlüsselschalter ab Ausgabestand 7

Im Gegensatz zu den bei früheren Ausgabeständen verfügbaren Schutzmechanismen für Flash Memory, müssen Sie hier erst den Schlüsselschalter über den Parameter "**RUN**/**STOP**-Schlüsselschalter" im Konfigurationsmenü der CPU freigeben, damit die CPU über die hier beschriebenen erweiterten Steuerfunktionen verfügen kann.

Beim Übergang der SPS in **RUN**-Modus verfügt der Schlüsselschalters über die gleichen Schutz- und Prüfmechanismen wie bisher. Das heisst, die SPS kann über den Schlüsselschalter nicht in **RUN**-Modus geschaltet werden, wenn sie in der Betriebsart **STOP**/**FEHLER** ist. Sie können in der Betriebsart **STOP/FEHLER** aber nicht fatale Fehler löschen und die SPS dann über den Schlüsselschalter in **RUN**-Modus schalten.

Enthalten die Fehlertabellen ***nicht fatale*** Fehler (dies sind Fehler, die die CPU nicht in den Modus **STOP**/**FEHLER** versetzen), geht die CPU in den **RUN**-Modus, wenn Sie den Schlüsselschalter das erste Mal von STOP auf RUN schalten. Die Fehlertabellen werden NICHT gelöscht.

Enthalten die Fehlertabellen ***fatale*** Fehler (CPU in **STOP**/**FEHLER**-Modus), beginnt die **RUN**-Anzeige auf der CPU beim ersten Umschalten des Schlüsselschalters von **STOP** auf **RUN** mit einer Frequenz von 2 Hz zu blinken. Gleichzeitig wird ein 5-Sekunden-Zeitgeber angestoßen. Die blinkende **RUN** -Anzeige meldet, dass die Fehlertabellen fatale Fehler enthalten. In diesem Fall geht die CPU nicht in den **RUN**-Zustand, selbst wenn der Schlüsselschalter auf **RUN** steht.

Fehlertabelle mit Schlüsselschalter löschen

Wenn Sie den Schlüsselschalter innerhalb der 5 Sekunden, die die **RUN**-Anzeige blinkt, von **RUN** auf **STOP** und zurück auf **RUN** schalten, werden die Fehler gelöscht und die CPU geht in den **RUN**-Modus. Die Anzeige hört auf zu blinken und leuchtet jetzt dauernd. Der Schalter muss hierbei aber mindestens 1/2 Sekunde lang in der Stellung **RUN** oder **STOP** gehalten werden.

Hinweis

Ist der 5-Sekunden-Zeitgeber abgelaufen (die **RUN**-Anzeige blinkt nicht mehr), bleibt die CPU im Zustand **STOP**/**FEHLER** mit Fehlern in der Fehlertabelle. Schalten Sie dann den Schlüsselschalter wieder von **STOP** auf **RUN**, wird der Prozess wiederholt und dies ist der erste Übergang.

Die nachstehende Tabelle gibt eine Zusammenfassung, wie sich die Einstellungen der beiden auf den Schlüsselschalter wirkenden CPU-Parameter (R/S-Schalter und IOScan-Stop) sowie die Einstellung des Schlüsselschalters auf die SPS auswirken.

Parameter R/S-Schalter in CPU-Konfiguration	Stellung des Schlüssel-schalters	Parameter IOScan-Stop in CPU-Konfiguration	SPS-Betrieb
AUS	X	X	Alle SPS-Programmiergeräte-Betriebsarten erlaubt.
EIN	EIN/RUN	X	Alle SPS-Programmiergeräte-Betriebsarten erlaubt.
EIN	AUS/STOP	X	SPS kann nicht auf RUN gehen.
EIN	Schlüsselschalter umschalten von OFF/STOP auf ON/RUN.	X	CPU geht nach RUN, wenn keine fatalen Fehler vorliegen. Andernfalls blinkt die RUN LED 5 Sekunden lang.
EIN	Schlüsselschalter umschalten von ON/RUN auf OFF/STOP.	Nein	SPS geht auf STOP–NO IO
EIN	Schlüsselschalter umschalten von ON/RUN auf OFF/STOP.	Ja	SPS geht auf STOP–IO

X = irrelevant

Erweiterter Speicherschutz bei CPUs ab Ausgabestand 8

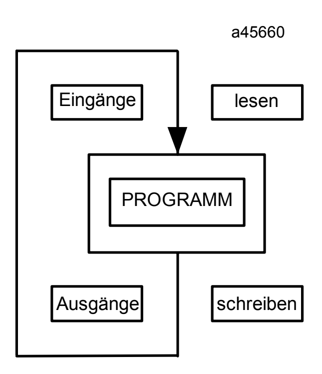
Bei CPUs ab Ausgabestand 8 kann der Schlüsselschalter, zusätzlich zu den zuvor besprochenen Eigenschaften, zum Schutz des RAM eingesetzt werden, so dass der RAM über die Programmiersoftware nicht verändert werden kann. Hierzu muss im Programmierpaket ein Parameter entsprechend eingestellt werden. Bei freigeschaltetem Speicherschutz sind zwei Operationstypen gesperrt: Anwenderprogramm und Konfiguration können nicht verändert werden und E/A dürfen nicht gesetzt oder gefordert werden. Diese Sperre wird bei den CPU-Modellen 35x und 36x über das Speicherschutzfeld im Logicmaster-Modulkonfigurationsmenü oder über das Speicherschutzfeld in der Karteikarte "Einstellungen" bei der Hardwarekonfiguration unter der VersaPro- oder Control-Software eingestellt. Die Standardeinstellung ist "gesperrt".

Abschnitt 2: Programmorganisation und Anwenderreferenzen/-daten

Die nachstehende Tabelle zeigt die bei den SPS der Series 90-30 möglichen Programmgrößen.

Modelle	Programmspeicher (kByte)
CPU311	6
CPU313, CPU323	12
CPU331	16
CPU340	32
CPU341	80
CPU350	80 (ab Ausgabestand 9) 32 (bis Ausgabestand 9)
CPU351, CPU352, CPU360, CPU363, CPU364	240 (ab Ausgabestand 9) 80 (vor Ausgabestand 9)

Bei den CPUs ab Ausgabestand 9 sind einige Speichergrößen der Modelle 351, 352 und 36x konfigurierbar (ausführliche Anweisungen sowie eine Erläuterung der verfügbaren Speichergrößen finden Sie in "Konfigurierbarer Speicher bei CPUs ab Modell 351” in Kapitel 10, Abschnitt 3 von *Logicmaster 90™ Series 90™-30/20/Micro Programmiersoftware, Anwenderhandbuch* (GFK-0466K oder höher). Bei einer CPU Modell 211 der SPS Series 90-20 ist die maximale Programmgröße 2 kB. Das Anwenderprogramm enthält Logik, die beim Hochlaufen verwendet wird. Maximal sind pro Programmblock (Haupt- oder Unterprogramm) 3000 Strompfade zulässig. Bei der SPS 90-30 beträgt die maximale Blockgröße 80 kByte für C-Blöcke und 16 kB für Kontaktplan- oder SFC-Blöcke. In SFC-Blöcken werden aber einige der 16 kB für den internen Datenblock verwendet. Die Logik wird von der SPS zyklisch bearbeitet.



Eine Liste der bei den einzelnen CPU-Modellen möglichen Programmgrößen und Referenzbereiche finden Sie in *SPS Series 90-30, Anwenderhandbuch,* GFK-0356 und in *SPS Series 90-20, Anwenderhandbuch*, GFK-0551.

Alle Programme besitzen eine Variablenvereinbarungstabelle, in der die im Anwenderprogramm zugewiesenen Variablen und Referenzbeschreibungen aufgelistet sind.

Mit dem Blockdeklarationseditor können im Hauptprogramm vereinbarte Unterprogrammblöcke aufgelistet werden.

Unterprogrammblöcke

Ein Programm kann während der Bearbeitung Unterprogramme aufrufen. Ehe ein Unterprogramm mit einem Befehl CALL aufgerufen werden kann, muss es über den Blockeditor deklariert werden. In jedem Programmblock sind bis zu 64 Unterprogrammblockdeklaratione und 64 CALL-Befehle möglich. Die Maximalgröße eines Unterprogramms beträgt 16 kB oder 3000 Strompfade, wobei Hauptprogramm und alle Unterprogramme die durch das CPU-Modell vorgegebenen Beschränkungen einhalten müssen.

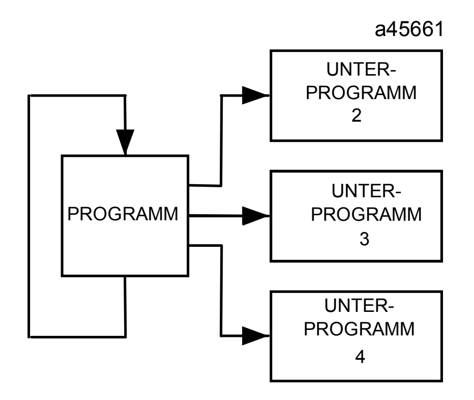
Hinweis

Weder bei der Series 90-20 noch bei der Micro gibt es Unterprogrammblöcke.

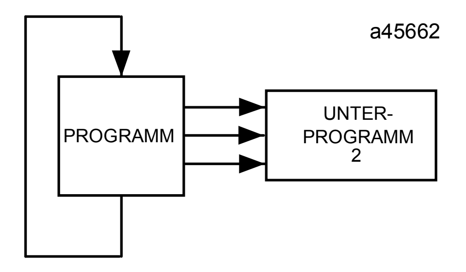
Der Einsatz von Unterprogrammen erfolgt wahlweise. Die Aufteilung eines Programms in kleinere Unterprogramme kann die Programmierung vereinfachen und das Programm verkleinern.

Beispiele von Unterprogrammblöcken

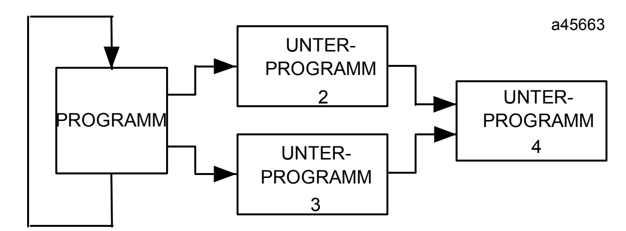
Die Logik eines Programms kann zum Beispiel in drei Unterprogramme aufgeteilt werden. Jedes dieser Unterprogramme kann dann vom Programm entsprechend den Anforderungen aufgerufen werden. In diesem Beispiel kann das Programm nur soviel Logik enthalten, wie zur Steuerung der Unterprogrammabfolge erforderlich ist.



Während des Programmablaufs kann ein Unterprogramm immer wieder aufgerufen werden. Programmteile, die in einem Programm mehrmals ausgeführt werden müssen, sollten daher in einem Unterprogrammblock stehen, der dann jeweils aufgerufen werden kann. Hierdurch wird die Programmlänge reduziert.



Unterprogrammblöcke können sowohl vom Hauptprogramm als auch von anderen Unterprogrammblöcken aufgerufen werden. Ein Unterprogrammblock kann sich sogar selbst aufrufen.



Die SPS trägt einen Fehler "Anwenderstack-Überlauf" ein und geht in die Betriebsart **STOP/FEHLER**, wenn mehr als acht verschachtelte Aufrufe gemacht werden. Bei der Zählung der Verschachtelungsebenen gilt das Programm als Ebene 1.

Wie Unterprogrammblöcke aufgerufen werden

Ein Unterprogrammblock wird ausgeführt, wenn er von der Programmlogik im Hauptprogramm oder von einem anderen Block aufgerufen wird.

```
|
|%I0004                                                          %T0001
|——| |——————————————————————————————————————————————————————————————( )—
|           _______________
|%I0006    |               |
|——| |—————|  CALL ASTRO   |—
|          |  (UNTERPRG.)  |
|          |_______________|
|
|%I0003  %I0010                                                  %Q0010
|——| |—————| |——————————————————————————————————————————————————————( )—
|
```

In diesem Beispiel sehen Sie den Befehl CALL, wie er im aufrufenden Block erscheint.

Periodische Unterprogramme

Ab Version 4.20 werden bei CPUs ab Modell 340 aufwärts periodische Unterprogramme unterstützt. Beachten Sie bitte folgende Einschränkungen:

1. Zeitfunktionsblöcke (TMR, ONDTR und OFDTR) werden in einem periodischen Unterprogramm nicht korrekt ausgeführt. Ein DOIO-Funktionsblock in einem periodischen Unterprogramm, zu dessen Referenzbereich Referenzen gehören, die einem intelligenten E/A-Modul (HSC, Power Mate APM, Genius, usw.) zugeordnet sind, bewirkt, dass die CPU die Datenverbindung zu dem Modul verliert. Die Kontakte FST_SCN und LST_SCN (%S1 und %S2) nehmen bei der Ausführung des periodischen Unterprogramms einen unbestimmten Wert an. Ein periodisches Unterprogramm kann keine anderen Unterprogramme aufrufen oder von diesen aufgerufen werden.

2. Befindet sich im Hauptchassis kein PCM-, CMM- oder ADC-Modul, beträgt die Latenzzeit (d.h. die maximale Zeitspanne zwischen dem Zeitpunkt, zu dem das periodische Unterprogramm ablaufen sollte und dem Zeitpunkt, zu dem es tatsächlich abläuft) des periodischen Unterprogramms etwa 0,35 Millisekunden. Befindet sich ein PCM-, CMM- oder ADC-Modul im Hauptchassis — selbst wenn es nicht konfiguriert ist oder benutzt wird — kann die Latenzzeit bis zu 2,25 Millisekunden betragen. Aus diesem Grund wird der Einsatz von periodischen Unterprogrammen bei Produkten auf PCM-Basis ***nicht*** empfohlen.

Anwenderreferenzen

Die in einem Anwenderprogramm verwendeten Daten werden entweder als diskrete Referenzen oder als Registerreferenzen gespeichert.

Tabelle 2-4. Registerreferenzen

Typ	Beschreibung
%R	Mit dem Präfix %R werden System-Registerreferenzen zugewiesen, in die Programmdaten abgespeichert werden können (z.B. Berechnungsergebnisse).
%AI	Das Präfix %AI steht für ein analoges Eingangsregister. Ihm folgt die Registeradresse der Referenz (z.B. %AI0015). In einem analogen Eingangsregister steht der Wert eines Analogeingangs oder ein anderer Wert.
%AQ	Das Präfix %AQ steht für ein analoges Ausgangsregister. Ihm folgt die Registeradresse der Referenz (z.B. %AQ0056). In einem analogen Ausgangsregister steht der Wert eines Analogausgangs oder ein anderer Wert.

Hinweis

Sämtliche Registerreferenzen bleiben erhalten, wenn die Versorgungsspannung der CPU ab- und wieder eingeschaltet wird.

Tabelle 2-5. Diskrete Referenzen

Typ	Beschreibung
%I	Das Präfix %I steht für diskrete Eingänge. Ihm folgt die Referenzadresse in der Eingangstabelle (z.B. %I00121). Die Referenzen %I liegen in der Eingangs-Statustabelle, in der die Zustände sämtlicher Eingangssignale gespeichert sind, die von den Eingangsmodulen im letzten Eingabezyklus empfangen wurden. Den diskreten Eingangsmodulen wird mit der Konfigurationssoftware oder dem Hand-Programmiergerät eine Referenzadresse zugeordnet. Solange keine Referenzadresse zugeordnet wurde, werden von diesem Modul keine Daten empfangen. Die %I-Daten können remanent oder nicht remanent sein.
%Q	Das Präfix %Q steht für physikalische diskrete Ausgänge. Die Spulenüberprüfungsfunktion der Logicmaster 90-30/20/Micro Software überprüft die %Q-Referenzen auf Mehrfachverwendung. Ab dem Software-Ausgabestand 3 können Sie die Ebene der Spulenüberprüfung einstellen (`EINFACH`, `WARNE MEHRFACH` oder `MEHRFACH`). Weitere Informationen zu dieser Funktion finden Sie im Programmiersoftware-Anwenderhandbuch, GFK-0466. Dem Präfix %Q folgt die Referenzadresse in der Ausgangstabelle (z.B. %IQ00016). Die Referenzen %Q liegen in der Ausgangs-Zustandstabelle, in der die Zustände der Ausgangsreferenzen so gespeichert sind, wie sie zuletzt vom Anwenderprogramm eingestellt wurden. Die Werte der Ausgangs-Zustandstabelle werden während des Ausgabezyklus an die Ausgangsmodule übertragen. Den diskreten Ausgangsmodulen wird mit der Konfigurationssoftware oder dem Hand-Programmiergerät eine Referenzadresse zugeordnet. Solange keine Referenzadresse zugeordnet wurde, werden zu diesem Modul keine Daten übertragen. Eine bestimmte %Q-Referenz kann remanent oder nicht remanent sein. *
%M	Das Präfix %M steht für eine interne Referenz. Die Spulenüberprüfungsfunktion der Logicmaster 90-30/20/Micro Software überprüft die %Q-Referenzen auf Mehrfachverwendung. Ab dem Software-Ausgabestand 3 können Sie die Ebene der Spulenüberprüfung einstellen (`EINFACH`, `WARNE MEHRFACH` oder `MEHRFACH`). Weitere Informationen zu dieser Funktion finden Sie in GFK-0466. Eine bestimmte %M-Referenz kann remanent oder nicht remanent sein. *

* Die Remanenz hängt vom Spulentyp ab. Weitere Informationen hierzu finden Sie unter "Datenremanenz" auf Seite 2-22.

Tabelle 2-5. Diskrete Referenzen - Fortsetzung

Typ	Beschreibung
%T	Das Präfix %T steht für eine diskrete temporäre Referenz. Diese Referenzen werden nicht auf Mehrfachverwendung überprüft und können daher im gleichen Programm selbst dann mehrfach eingesetzt werden, wenn die Spulenverwendungs-Überprüfungsfunktion aktiviert ist. %T kann zur Vermeidung von Adressierungskonflikten mit den Funktionen Cut/Paste [Einfügen/Ausfügen] und Schreiben und Einfügen von Dateien verwendet werden. Da dieser Speicherbereich nur für den temporären Gebrauch vorgesehen ist, kann er nicht mit remanenten Merkern verwendet werden. Der Inhalt geht bei Spannungsausfall und beim Übergang **RUN-STOP-RUN** verloren.
%S	Das Präfix %S steht für Systemreferenzen. Mit diesen Referenzen kann auf spezielle SPS-Daten zugegriffen werden (z.B. Zeitglieder, Zyklusdaten, Fehlerdaten). Zu den Systemreferenzen gehören die Referenzen %S, %SA, %SB und %SC. %S, %SA, %SB und %SC können für beliebige logische Kontakte verwendet werden. %SA, %SB und %SC können für remanente Spulen –(M)– verwendet werden. %S kann als Wort- oder Bitfolge-Eingangsargument in Funktionen oder Funktionsblöcken verwendet werden. %SA, %SB und %SC können als Wort- oder Bitfolge-Ein- oder Ausgangsargument in Funktionen und Funktionsblöcken verwendet werden.
%G	Das Präfix %G steht für Globalreferenzen. Über diese Referenzen kann auf Daten zugegriffen werden, die von mehreren SPS-System gemeinsam benutzt werden. %G-Referenzen können bei Kontakten und remanenten Spulen verwendet werden, da der %G-Speicher immer remanent ist. %G kann nicht mit nicht-remanenten Spulen verwendet werden.

Transitionen und Overrides

Den Anwenderreferenzen %I, %Q, %M und %G sind Transitions- und Overridebits zugeordnet. Die Referenzen %T, %S, %SA, %SB, %SC besitzen Transitionsbits, jedoch keine Overridebits. Die CPU verwendet Transitionsbits für Zähler und Übergangsmerker. Beachten Sie, dass Zähler eine andere Art Transitionsbits verwenden, als dies bei Spulen der Fall ist. Transitionsbits für Zähler werden in der einrichtenden Referenz gespeichert.

In CPUs ab Modell 331 aufwärts können Overridebits gesetzt werden. Bei gesetzten Overridebits können die zugehörigen Referenzen nur über das Programmiergerät verändert werden, nicht vom Programm oder Eingabegerät. Weder die CPU-Modell 323, 321, 313 311 noch die Micro-CPUs unterstützen diskrete Override-Referenzen.

Datenremanenz

Daten sind "remanent", wenn sie beim Anhalten der SPS gerettet werden. Bei der SPS Series 90-30 werden erhalten Programm, Fehlertabellen und Diagnosedaten, Überspeicherungen (Override) und Fixierung von Ausgängen, Wortdaten (%R, %AI, %AQ), Bitdaten (%I, %SC, %G, Fehlerbits und reservierte Bits), %Q- und %M-Daten (sofern sie nicht mit nicht-remanenten Merkern verwendet werden), sowie Wortdaten, die in %M und %Q gespeichert sind. %T-Daten werden nicht gepuffert. Obwohl %SC remanent ist, sind %S, %SA und %SB in der Grundeinstellung nicht remanent.

%Q- und %M-Referenzen sind nicht remanent (d.h. sie werden beim Einschalten gelöscht, wenn die SPS von **STOP** auf **RUN** umschaltet), wenn Sie mit nicht remanenten Merkern zusammen verwendet werden. Zu den nicht remanenten Merkern gehören Spulen —( )—, negierte Spulen —(/)—, SET-Spulen —(S)— und RESET-Spulen —(R)—.

Werden %Q- und %M-Referenzen zusammen mit remanenten Merkern oder als Funktionsblockausgang verwendet, wird ihr Inhalt über einen Ausfall der Versorgungsspannung hinweg und beim

Übergang **RUN-STOP-RUN** erhalten. Zu den remanenten Merkern gehören remanente Merker —(M)—, negierte remanente Merker —(/M)—, remanente SET-Spulen —(SM)— und remanente RESET-Spulen —(RM)—.

Der Remanenzzustand (remanent oder nicht remanent) einer %Q- oder %M-Referenz wird auf der Grundlage des Merkertyps durch die letzte Programmierung dieser %Q- oder %M-Referenz mit einer Merkeranweisung festgelegt. Wurde zum Beispiel %Q0001 zuletzt als Referenz einer remanenten Spule programmiert, sind die Daten von %Q0001 remanent. Wurde jedoch %Q0001 zuletzt als Referenz einer nicht remanenten Spule programmiert, sind die Daten von %Q0001 nicht remanent.

Datentypen

Tabelle 2-6. Datentypen

Typ	Bezeich-nung	Beschreibung	Datenformat
INT	Ganze Zahl mit Vorzeichen	Dieser Datentyp belegt 16 Datenbits im Speicher und wird im Zweierkomplement dargestellt. Der zulässige Bereich liegt zwischen –32.768 und +32.767.	Register 1 S | (16 Bitpositionen) 16 1
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen	Dieser Datentyp belegt 32 Datenbits im Speicher (zwei aufeinanderfolgende 16-Bit-Speicherplätze) und wird im Zweierkomplement dargestellt (Bit 32 ist das Vorzeichenbit). Der zulässige Bereich liegt zwischen –2.147.483.648 und +2.147.483.647.	Register 2 Register 1 S | 32 17 16 1 (Wert im Zweierkomplement)
BIT	Bit	Dieser Datentyp ist die kleinste Einheit im Speicher. Er besitzt zwei Zustände, 1 oder 0. Eine Bitfolge kann die Länge N haben.	
BYTE	Byte	Dieser Datentyp hat einen 8-Bit-Wert. Der zulässige Bereich liegt zwischen 0 und 255 (0 bis FF in Hexadezimaldarstellung).	
WORD	Wort	Dieser Datentyp belegt 16 aufeinanderfolgende Bits im Datenspeicher. Diese Bits stellen aber keine Zahl dar, sondern sind unabhängig voneinander. Jedes Bit wird für sich betrachtet und stellt einen eigenen Binärzustand (0 oder 1) dar. Der zulässige Bereich eines Wortwertes liegt zwischen 0 und FFFF.	Register 1 (16 Bitpositionen) 16 1
DWORD	Doppelwort	Dieser Datentyp besitzt die gleichen Eigenschaften wie das Einzelwort. Er belegt jedoch 32 aufeinanderfolgende Bits im Datenspeicher (anstelle von 16 beim Einzelwort).	Register 2 Register 1 32 17 16 1 (32 Bitzustände)
BCD-4	4-stellige BCD-Zahl	Vierstellige BCD-Zahlen belegen 16 Bits im Datenspeicher. Jede BCD-Stelle belegt vier Bits und kann eine Zahl zwischen 0 und 9 darstellen. Die BCD-Codierung der 16 Bits ergibt einen Zahlenbereich zwischen 0 und 9999.	Register 1 4 | 3 | 2 | 1 (4 BCD-Stellen 16 13 9 5 1
REAL	Fliesskommazahl	Dieser Datentyp belegt 32 aufeinanderfolgende Bits im Speicher (zwei aufeinanderfolgende 16-Bit-Speicherplätze). Der zulässige Bereich liegt zwischen ± 1,401298E-45 und ± 3,402823E+38.	Register 2 Register 1 S | 32 17 16 1 (Wert im Zweierkomplement)

S = Vorzeichenbit (0 = positiv, 1 = negativ).

System-Statusreferenzen

Die System-Statusreferenzen der SPS Series 90 sind den %S-, %SA-, %SB- und %SC-Speichern zugeordnet. Sie besitzen jeweils eine symbolische Adresse. Beispiele für Zeitreferenzen sind T_10MS, T_100MS, T_SEC und T_MIN. Beispiele von Funktionsreferenzen sind FST_SCN, ALW_ON und ALW_OFF.

Hinweis

%S-Bits können nur gelesen werden. %SA-, %SB- oder %SC-Bits können gelesen und verändert werden.

In der nachstehenden Tabelle sind alle in einem Anwenderprogramm möglichen Systemreferenzen zusammengefasst. Bei der Eingabe eines Programms können entweder die Referenz oder die symbolische Adresse verwendet werden. In Kapitel 3 "Fehlerbeschreibungen und Fehlerbehebung" werden Fehler und Fehlerbehebung ausführlicher beschrieben.

Sie können diese speziellen Namen nicht in einem anderen Zusammenhang benutzen.

Tabelle 2-7. System-Statusreferenzen

Referenz	Symbolische Adresse	Definition
%S0001	FST_SCN	Wird auf 1 gesetzt, wenn der aktuelle Zyklus der erste Zyklus ist.
%S0002	LST_SCN	Wird von 1 auf 0 rückgesetzt, wenn der aktuelle Zyklus der letzte Zyklus ist.
%S0003	T_10MS	Zeitgeberkontakt, 0,01 s
%S0004	T_100MS	Zeitgeberkontakt, 0,1 s
%S0005	T_SEC	Zeitgeberkontakt, 1,0 s
%S0006	T_MIN	Zeitgeberkontakt, 1,0 Minute
%S0007	ALW_ON	Immer EIN
%S0008	ALW_OFF	Immer AUS.
%S0009	SY_FULL	Wird gesetzt, wenn die SPS-Fehlertabelle voll wird. Wird gelöscht, wenn ein Eintrag aus der SPS-Fehlertabelle entfernt wurde oder wenn die SPS-Fehlertabelle gelöscht wurde.

Tabelle 2-7. System-Statusreferenz - Fortsetzung

Referenz	Bezeich-nung	Definition
%S0010	IO_FULL	Wird gesetzt, wenn die E/A-Fehlertabelle voll wird. Wird gelöscht, wenn ein Eintrag aus der E/A-Fehlertabelle entfernt wird oder die E/A-Fehlertabelle gelöscht wird.
%S0011	OVR_PRE	Wird gesetzt, wenn im %I-, %Q-, %M- oder %G-Speicher eine Referenz geforced ist besteht.
%S0013	PRG_CHK	Wird gesetzt, wenn die Hintergrundprogrammüberprüfung aktiv ist.
%S0014	PLC_BAT	Wird gesetzt, um bei einer CPU ab Ausgabestand 4 eine niedrige Batterie-spannung anzuzeigen. Die Kontaktreferenz wird einmal pro Zyklus aktua-lisiert.
%S0017	SNPXACT	SNP-X-Host ist aktiv an die CPU angeschlossen.
%S0018	SNPX_RD	SNP-X-Host hat Daten von der CPU gelesen.
%S0019	SNPX_WT	SNP-X-Host hat Daten zur CPU geschrieben.
%S0020		Wird gesetzt, wenn eine Vergleichsfunktion mit REAL-Daten erfolgreich ausgeführt wird. Wird gelöscht, wenn einer der Eingänge keine Zahl ist.
%S0032		Reserviert für Logicmaster 90-30/20/Micro Software.
%SA0001	PB_SUM	Wird gesetzt, wenn die für das Programm berechnete Prüfsumme nicht mit der Referenz-Prüfsumme übereinstimmt. War die Ursache ein vorübergehender Fehler, kann das Bit gelöscht wer-den, indem das Programm erneut in die CPU gespeichert wird. War die Ursache ein RAM-Fehler, muss die CPU ersetzt werden.
%SA0002	OV_SWP	Wird gesetzt, wenn die SPS erkennt, dass der letzte Zyklus länger als die vom Anwender vorgegebene Zeit war. Wird gelöscht, wenn die SPS er-kennt, dass der letzte Zyklus nicht länger als die vom Anwender angegebe-ne Zeit war. Wird auch beim Übergang von **STOP** nach **RUN** gelöscht. Nur zulässig, wenn die SPS im Modus **KONSTANTE ZYKLUSDAUER** ist.
%SA0003	APL_FLT	Wird gesetzt, wenn im Anwenderprogramm ein Fehler auftritt. Wird beim Übergang der SPS von **STOP** nach **RUN** gelöscht.
%SA0009	CFG_MM	Wird gesetzt, wenn beim Einschalten oder Speichern der Konfiguration eine Diskrepanz in der Konfiguration festgestellt wird. Wird beim Einschalten der SPS gelöscht, wenn keine Diskrepanz mehr festgestellt wird oder beim Speichern einer Konfiguration, die mit der Hardware übereinstimmt.
%SA0010	HRD_CPU	Wird gesetzt, wenn die Diagnosefunktionen ein Problem mit der CPU-Hardware erkennen. Wird gelöscht, wenn das CPU-Modul ausgewechselt wird.
%SA0011	LOW_BAT	Wird gesetzt, wenn die Batteriespannung abfällt. Wird gelöscht, wenn die Batterie ausgewechselt und die CPU mit normaler Batteriespannung wieder eingeschaltet wird.
%SA0014	LOS_IOM	Wird gesetzt, wenn die Kommunikation zwischen einem E/A-Modul und der SPS CPU ausfällt. Wird gelöscht, indem das Modul ausgewechselt und die Spannung zum Hauptchassis aus- und wieder eingeschaltet wird.
%SA0015	LOS_SIO	Wird gesetzt, wenn die Kommunikation zwischen einem Zusatzmodul und der SPS CPU ausfällt. Wird gelöscht, indem das Modul ersetzt und die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird.

Tabelle 2-7. System-Statusreferenzen - Fortsetzung

Referenz	Symbolische Adresse	Definition
%SA0019	ADD_IOM	Wird gesetzt, wenn ein E/A-Modul in einem Chassis hinzugefügt wird. Wird gelöscht, indem die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird, nachdem die Konfigurationsdatei aktualisiert wurde.
%SA0020	ADD_SIO	Wird gesetzt, wenn ein Zusatzmodul in einem Chassis hinzugefügt wird. Wird gelöscht, indem die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird, nachdem die Konfigurationsdatei aktualisiert wurde.
%SA0027	HRD_SIO	Wird gesetzt, wenn in einem Zusatzmodul ein Hardwarefehler erkannt wird. Wird gelöscht, wenn das Modul ausgewechselt und die Spannung zum Hauptchassis aus- und wieder eingeschaltet wird.
%SA0031	SFT_SIO	Wird gesetzt, wenn in einem Zusatzmodul ein nicht behebbarer Softwarefehler erkannt wird. Wird gelöscht, indem die Versorgungsspannung zum Hauptchassis aus- und wieder eingeschaltet wird, nachdem die Konfigurationsdatei aktualisiert wurde.
%SB0010	BAD_RAM	Wird gesetzt, wenn die CPU beim Einschalten einen RAM-Fehler erkennt. Wird gelöscht, wenn die CPU beim Einschalten erkennt, dass der RAM fehlerfrei ist.
%SB0011	BAD_PWD	Wird gesetzt, wenn ein durch Passwort geschützter Zugriff verletzt wird. Wird gelöscht, wenn die SPS-Fehlertabelle gelöscht wird.
%SB0013	SFT_CPU	Wird gesetzt, wenn die CPU einen nicht behebbaren Fehler in der Software erkennt. Wird gelöscht, wenn die SPS-Fehlertabelle gelöscht wird.
%SB0014	STOR_ER	Wird gesetzt, wenn während des Abspeicherns ins Programmiergerät ein Fehler auftritt. Wird gelöscht, wenn ein Speichervorgang erfolgreich beendet wurde.
%SC0009	ANY_FLT	Wird gesetzt, wenn ein beliebiger Fehler auftritt. Wird gelöscht, wenn beide Fehlertabellen leer sind.
%SC0010	SY_FLT	Wird gesetzt, wenn ein Fehler auftritt, der einen Eintrag in die SPS-Fehlertabelle verursacht. Wird gelöscht, wenn die SPS-Fehlertabelle leer ist.
%SC0011	IO_FLT	Wird gesetzt, wenn ein Fehler auftritt, der einen Eintrag in die E/A-Fehlertabelle verursacht. Wird gelöscht, wenn die E/A-Fehlertabelle leer ist.
%SC0012	SY_PRES	Wird gesetzt, solange die SPS-Fehlertabelle noch mindestens einen Eintrag enthält. Wird gelöscht, wenn die SPS-Fehlertabelle leer ist.
%SC0013	IO_PRES	Wird gesetzt, solange die E/A-Fehlertabelle noch mindestens einen Eintrag enthält. Wird gelöscht, wenn die E/A-Fehlertabelle leer ist.
%SC0014	HRD_FLT	Wird gesetzt, wenn ein Hardwarefehler auftritt. Wird gelöscht, wenn beide Fehlertabellen leer sind.
%SC0015	SFT_FLT	Wird gesetzt, wenn ein Softwarefehler auftritt. Wird gelöscht, wenn beide Fehlertabellen leer sind.

Hinweis: Alle hier nicht aufgeführten %S-Referenzen sind reserviert und dürfen im Programm nicht verwendet werden.

Funktionsblockstruktur

Jeder Strompfad besteht aus einer oder mehreren Programmieranweisungen. Hierbei kann es sich um einfache Relais oder um komplexere Funktionen handeln.

Format der Kontaktplanrelais

Die Programmiersoftware verfügt über mehrere Arten von Relaisfunktionen, die grundlegend für Stromfluss und logische Verknüpfungen im Programm sorgen. Zu ihnen gehören ein Schließerkontakt und eine negierte Spule. Jeder Relaiskontakt und jede Spule besitzt einen Eingang und einen Ausgang. Zusammen ermöglichen sie den Stromfluss durch den Kontakt oder die Spule.

Jedem Relaiskontakt und jeder Spule muss eine Referenz zugeteilt werden, die bei der Auswahl des Relais eingegeben wird. Bei einem Kontakt stellt die Referenz einen Speicherplatz dar, der den Stromfluss in den Kontakt hinein festlegt. Ist im nachstehenden Beispiel die Referenz %I0122 EIN, fließt Strom durch diesen Relaiskontakt.

```
%I0122
 -| |-
```

Bei einer Spule stellt die Referenz den Speicherplatz dar, der durch den Stromfluss in die Spule kontrolliert wird. Fließt in diesem Beispiel der Strom in die linke Seite der Spule, wird die Referenz %Q0004 durchgeschaltet.

```
%Q0004
 -( )-
```

Sowohl die Programmiersoftware als auch das Hand-Programmiergerät verfügen über eine Spulenverwendungs-Überprüfungsfunktion, die feststellt, ob %Q- oder %M-Referenzen mit Relaisspulen oder Funktionsausgängen mehrfach benutzt wurden.

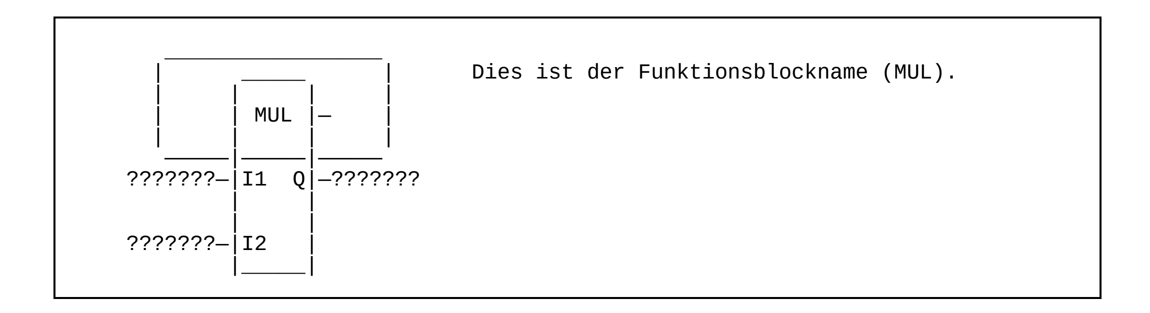
Format der Programm-Funktionsblöcke

Einige Funktionsblöcke sind sehr einfach aufgebaut. Hierzu gehört die MCR-Funktion, die mit dem abgekürzten Funktionsnamen in eckigen Klammern dargestellt wird:

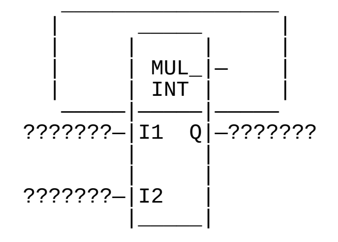
```
-[ MCR ]-
```

Andere Funktionen sind komplexer. Sie können die Eingabe der für die Funktion benötigten Informationen an mehreren Stellen erforderlich machen.

Nachstehend wird der allgemein verwendbare Funktionsblock Multiplikation (MUL) dargestellt. Die Parameter verändern sich mit der Art des Funktionsblocks. Seine Bestandteile sind typisch für zahlreiche Programmfunktionen. Der obere Teil des Funktionsblocks zeigt den Namen der Funktion. Er kann auch einen Datentyp anzeigen (hier: Ganze Zahl mit Vorzeichen).



Dies ist der Funktionsblockname (MUL).



Dies ist der Funktionsblockname (MUL) und Datentyp (INT). INT (ganze Zahl mit Vorzeichen) stellt Typ und Größe der Daten dar, die zu bearbeiten sind.

Bei vielen Programmfunktionen können Sie den Datentyp für die Funktion einstellen, nachdem Sie die Funktion ausgewählt haben. Den Datentyp der MUL-Funktion könnten Sie zum Beispiel auf doppeltgenaue ganze Zahl mit Vorzeichen abändern. Weitere Informationen zu den Datentypen finden Sie weiter vorne in diesem Kapitel.

Funktionsblockparameter

Jede Linie, die in die linke Seite eines Funktionsblocks führt, stellt einen Eingang zu dieser Funktion dar. Es gibt zwei Arten Eingänge in einen Funktionsblock: Konstanten und Referenzen. Eine Konstante ist ein expliziter Wert. Eine Referenz ist die Adresse eines Wertes.

Im nachstehenden Beispiel ist der Eingangsparameter I1 des ADD-Funktionsblocks eine Konstante, während Eingangsparameter I2 eine Referenz darstellt.

```
|          _____
|%I0001   |     |                                                          %Q0001
|——| |————| ADD_|———————————————————————————————————————————————————————————( )—
|         | INT |
|         |     |
| CONST —|I1  Q|—%R0002
| +00010  |     |
|         |     |
|%R0001 —|I2   |
|         |_____|
|
```

Jede Linie, die den Funktionsblock auf der rechten Seite verlässt, stellt einen Ausgang dar. Aus einem Funktionsblock gibt es nur eine Art Ausgang, die Referenz. Ausgangswerte können niemals in eine Konstante geschrieben werden.

Geben Sie bei den Fragezeichen auf der linken Seite des Funktionsblocks entweder den Datenwert selbst ein, oder die Referenzadresse, unter der der Datenwert gefunden werden kann, oder eine Variable, die die Referenzadresse darstellt, unter der der Datenwert gefunden werden kann. An Fragezeichen auf der rechten Seite eines Funktionsblocks geben Sie normalerweise eine Referenzadresse für die Daten an, die vom Funktionsblock ausgegeben werden, oder aber eine Variable, die die Referenzadresse für Daten darstellt, die von dem Funktionsblock ausgegeben werden.

```
                 _____
                |     |
               —| MUL |—
                |     |
      __________|     |__________
     | ???????—|I1  Q|—??????? |
     |          |     |__________
     |          |     |      |
     | ???????—|I2   |       ______  Dies ist der Ausgangsparameter (Q)
      __________|     |               für den Funktionsblock.
            |   |_____|
            |
            |______ Dies sind die Eingangsparameter (I1 und I2)
                    für den Funktionsblock.
```

```
                 _____
                |     |
               —| MUL_|—
                | INT |
      __________|     |__________
     | ???????—|I1  Q|—??????? |
     |          |     |__________
     |          |     |      |
     | ???????—|I2   |       ______  Dies ist der Ausgangsparameter (Q)
      __________|     |               für den Funktionsblock.
            |   |_____|
            |
            |______ Dies sind die Eingangsparameter (I1 und I2)
                    für den Funktionsblock.
```

Die meisten Funktionsblöcke verändern die Eingangsdaten nicht. Sie schreiben die Ergebnisse der Operation in eine Ausgangsreferenz.

Bei Funktionen, die Tabellen bearbeiten, kann für die Funktion ein Längenwert eingestellt werden. Im nachstehenden Funktionsblock gibt der Operand LEN an, wieviel Worte kopiert werden sollen.

```
                      _______
                     |       |
       (Freigabe) ---| MOVE_ |--- (ok)
                     |  WORD |
                     |       |
          ???????----|I1    Q|----???????
                     |       |
                     | LEN   |
                     | 00003 |
                     |_______|
```

Zeitgeber-, Zähler-, BITSEQ- und ID-Funktionen benötigen eine Adresse für die drei Worte im Speicher (Register), in denen der aktuelle Wert, der voreingestellte Wert und ein Steuerwort (Instanz) für die Funktion eingetragen werden.

```
                   ______
                  |      |
     (Freigabe) --|ONDTR |-- Q
                  |1.00s |
                  |      |
  (Rücksetzen) ---|R     |
                  |      |
                  |      |
       ???????----|PV    |
                  |______|
                 (Adresse)
```

Stromfluss zu und von einer Funktion

Der Strom fließt oben links in die Funktion. Häufig wird der Stromfluss zum Funktionsblock durch eine Freigabelogik gesteuert. Ist dies nicht der Fall, wird der Funktionsblock unbedingt in jedem CPU-Zyklus ausgeführt.

```
        Freigabelogik
              |
              |  Stromfluss in die Funktion
              |   |
              |   |          Stromfluss aus der Funktion heraus
              |   |   ______     |
     %I0001       |  |      |                       %Q0001
   ----| |---------| MUL_  |---------------------------( )--
                  | INT   |                             ^
                  |       |                             |
     %R0123 ------|I1    Q|---%R0124               Zeigt Zustand der
                  |       |                          Referenz an
                  |       |
       CONST -----|I2     |
       00002      |_______|
```

Hinweis

Funktionsblöcke können nicht direkt an die linke Stromschiene angeschlossen werden. Um eine Funktion in jedem Zyklus aufzurufen, können Sie %S7 verwenden, das Bit ALW_ON (immer EIN), mit einem an der Stromschiene angeschlossenen Schließerkontakt.

Der Strom fließt oben rechts aus der Funktion heraus. Er kann wahlweise zu anderen Programmelementen oder zu einer Spule geführt werden. Funktionsblöcke leiten den Stromfluss weiter, wenn sie erfolgreich ausgeführt wurden.

Abschnitt 3: Ein- und Abschaltsequenzen

Bei der SPS Series 90-30 sind zwei Einschaltsequenzen möglich: Kaltstart und Warmstart. Normalerweise benutzt die CPU die Kaltstartsequenz. Sind jedoch in einem SPS-System ab Modell 331 aufwärts zwischen Ausschalten und Wiedereinschalten weniger als fünf Sekunden verstrichen, wird die Warmstartsequenz verwendet.

Einschalten

Bei einem Kaltstart laufen nacheinander folgende Ereignisse ab: (Bei einem Warmstart wird Schritt 1 übersprungen).

1. Die CPU führt einen Selbsttest durch. Hierzu gehört die Überprüfung eines Teils des batteriegepufferten RAMs um festzustellen, ob der RAM gültige Daten enthält.
2. Gibt es ein EPROM, EEPROM oder Flash Memory und gibt die PROM-Einschaltoption an, dass der Inhalt des PROMs benutzt werden soll, wird der Inhalt des PROMs in den RAM kopiert. Gibt es kein EPROM, EEPROM oder Flash Memory, bleibt der RAM unverändert und wird nicht durch den Inhalt des PROMs überschrieben.
3. Die CPU fragt die einzelnen Steckplätze im System ab um festzustellen, welche Module vorhanden sind.
4. Die Hardwarekonfiguration wird nun mit der Softwarekonfiguration verglichen. Erkannte Diskrepanzen werden als Fehler bewertet und gemeldet. Entspricht ein eingebautes Modul nicht den Angaben in der Softwarekonfiguration, wird dies auch als Fehler bewertet und gemeldet.
5. Liegt keine Softwarekonfiguration vor, verwendet die CPU die Standardkonfiguration.
6. Die CPU richtet die Kommunikationskanäle zwischen ihr und den intelligenten Modulen ein.
7. Im letzten Schritt wird der Modus des ersten Zyklus auf der Grundlage der CPU-Konfiguration bestimmt. Bei **RUN** läuft der Zyklus so ab, wie unter "Übergang von **STOP** auf **RUN** " beschrieben. Abbildung 2-5 auf der nächsten Seite zeigt die Entscheidungsfolge für die CPU bei der Festlegung, ob aus dem PROM kopiert wird und ob der Hochlauf im Modus **STOP** oder **RUN** erfolgt.

Hinweis

Die Schritte 2 bis 7 gelten nicht für die SPS Series 90 Micro. Informationen zu den Ein- und Ausschaltsequenzen der Micro finden Sie in Kapitel 5 "Systembetrieb" von GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

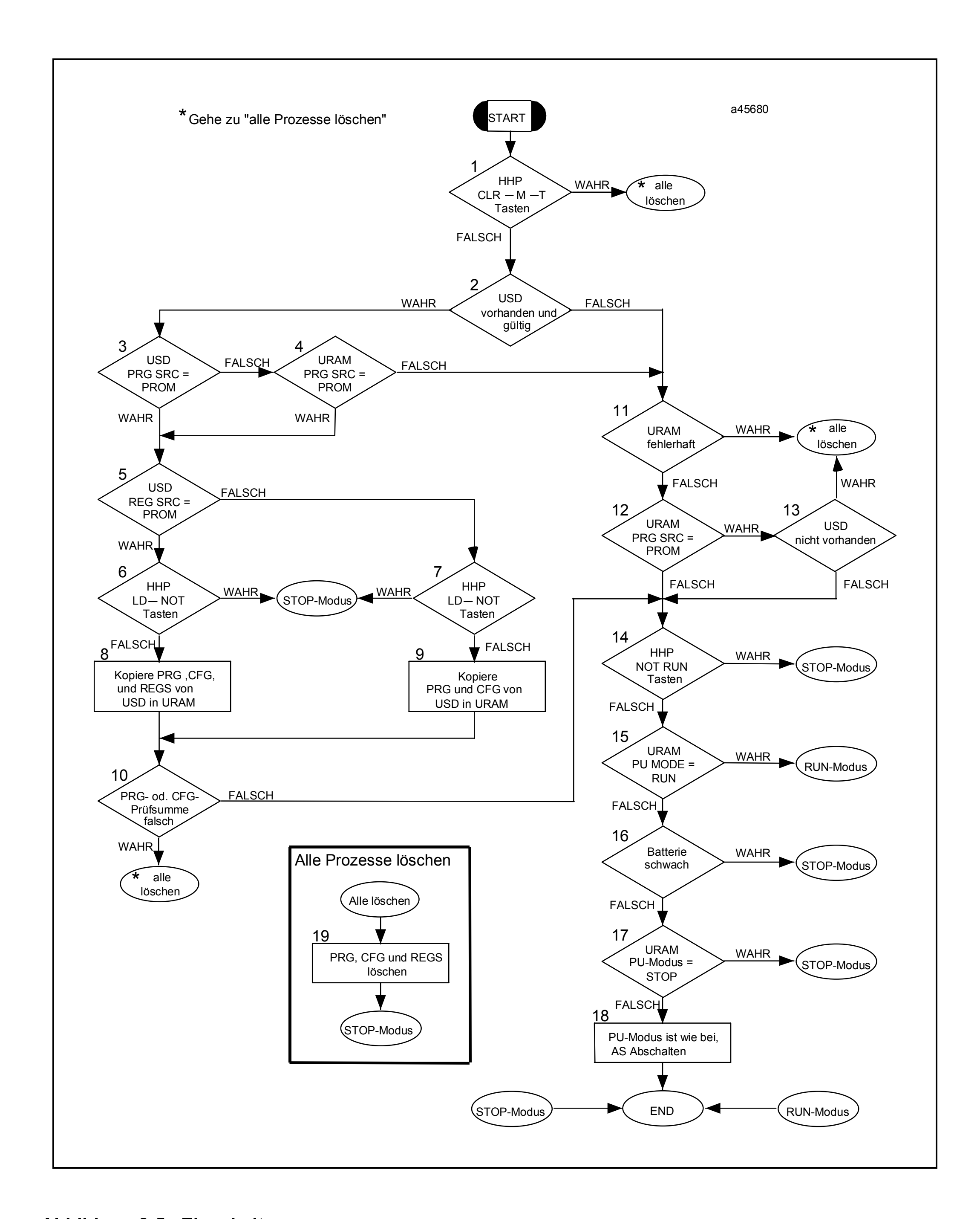


Abbildung 2-5. Einschaltsequenz

Vor der START-Anweisung auf dem Einschalt-Blockschaltbild durchläuft die CPU Einschalt-Diagnosefunktionen, die die verschiedenen von der CPU benutzten Peripheriegeräte testen, und prüft den RAM. Nach Abschluss dieser Diagnosefunktionen werden die internen Datenstrukturen sowie die von der CPU benutzten Peripheriegeräte initialisiert. Anschließend stellt die CPU fest, ob der Anwender-RAM fehlerhaft ist. Ist dies der Fall, werden Anwenderprogramm und Konfiguration gelöscht und auf Standardwerte gesetzt, und es werden alle Anwenderregister gelöscht.

BEGRIFFE AUS DEM BLOCKSCHALTBILD:

PRG = Anwenderprogramm

CFG = Anwenderkonfiguration

REGS = Anwenderregister (%I, %Q, %M, %G, %R, %AI, %AQ).

USD = Anwenderspeicher (EEPROM oder Flash Memory).

URAM = nichtflüchtiger Anwender-RAM, der PRG, CFG und REGS enthält.

ERLÄUTERUNGEN ZUM BLOCKSCHALTBILD:

(1) Wurden am HHP beim Einschalten die Tasten <CLR> und >M_T> gedrückt, um den gesamten URAM zu löschen?

(2) Ist der USD vorhanden (darf nur bei Modellen fehlen, die EEPROM benutzen) und enthält der USD gültige Daten?

(3) Wurde der Parameter PRG SRC im USD auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?

(4) Wurde der Parameter PRG SRC im URAM auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?

(5) Wurde der Parameter REG SRC im USD auf PROM eingestellt (d.h. REGS werden aus dem USD geladen)?

(6 & 7) Wurden am HHP beim Einschalten die Tasten <LD> und <NOT> gedrückt, um zu verhindern, dass PRG, CFG und REGS vom USD geladen werden?

(8) PRG, CFG und REGS vom USD in URAM kopieren.

(9) PRG und CFG vom USD in URAM kopieren.

(10) Sind die gerade vom USD geladenen Prüfsummen von PRG oder CFG ungültig?

(11) Ist der URAM verstümmelt? Mögliche Ursachen: Abschalten ohne oder mit schwacher Batterie, oder Firmware-Aktualisierung.

(12) Wurde der Parameter PRG SRC im URAM auf PROM eingestellt (d.h. PRG und CFG werden aus dem USD geladen)?

(13) Ist der USD vorhanden (nur bei Modellen, die EEPROM benutzen)?

(14) Wurden am HHP beim Einschalten die Tasten <**NOT**> und <**RUN**> gedrückt, um unbedingt im STOP-Modus hochzufahren?

(15) Wurde der Parameter PWR UP im URAM auf **RUN** gesetzt?

(16) Ist die Batteriespannung zu niedrig?

(17) Wurde der Parameter PWR UP im URAM auf **STOP** gesetzt?

(18) Einschaltmodus auf Ausschaltmodus setzen.

(19) PRG, CFG und REGS löschen.

Hinweis

Der erste Teil des Diagramms auf der vorhergehenden Seite gilt nicht für die SPS Series 90 Micro. Informationen zu den Ein- und Ausschaltsequenzen der Micro finden Sie in Kapitel 5 "Systembetrieb" von GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

Abschalten

Eine Systemabschaltung findet statt, wenn die Stromversorgung erkennt, dass die Eingangswechselspannung länger als eine Periode ausgefallen ist oder wenn die Ausgangsspannung der 5-V-Stromversorgung auf einen Wert von weniger als 4,9 VDC abgefallen ist.

Abschnitt 4: Uhren und Zeitgeber

Zu den in der SPS Series 90-30 enthaltenen Uhren und Zeitgebern gehören eine Betriebszeituhr, eine Tageszeituhr (Modelle 331, 340/341, 351/352 und 28-Punkt Micro), ein Überwachungszeitgeber (Watchdog) und ein Zeitgeber für konstante Zykluszeit. Außerdem gibt es noch drei Arten von Zeitgeber-Funktionsblöcken: eine Einschaltverzögerung, eine Abschaltverzögerung und eine remanente Einschaltverzögerungs. Vier zeitgesteuerte Kontakte schalten in Abständen von 0,01 Sekunden, 0,1 Sekunden, 1,0 Sekunde und 1 Minute.

Betriebszeituhr

Die Betriebszeituhr erfasst die seit dem Einschalten der CPU verstrichene Zeit in Schritten von 100 Mikrosekunden. Diese Uhr ist nicht nullspannungssicher und läuft bei jedem Einschalten von neuem an. Einmal pro Sekunde unterbricht die Hardware die CPU, damit ein Sekundenwert aufgezeichnet werden kann. Etwa 100 Jahre nach Beginn der Zeiterfassung fängt dieser Sekundenwert wieder bei Null an.

Da die Betriebszeituhr die Basis für die Operationen der Systemsoftware und Zeitgeber-Funktionsblöcke bildet, darf sie über das Programmiergerät oder vom Programm aus nicht rückgesetzt werden. Das Anwenderprogramm kann jedoch mit der SVCREQ-Funktion #16 den aktuellen Wert der Betriebszeituhr auslesen.

Zeituhr

Bei der 28-Punkt Micro und ab dem Modell 331 der SPS Series 90-30 wird die Tageszeit durch eine Tageszeituhr verwaltet. Diese Zeituhr umfasst sieben Zeitfunktionen:

- Jahr (zwei Stellen)
- Monat
- Tag
- Stunde
- Minute
- Sekunde
- Wochentag

Die Tageszeituhr ist batteriegepuffert und erhält ihren Zustand über einen Spannungsausfall hinweg. Solange die Uhr nicht initialisiert wurde, sind die in ihr enthaltenen Werte jedoch ohne Bedeutung. Das Anwenderprogramm kann mit der SVCREQ-Funktion #7 die Tageszeituhr lesen und einstellen. Die Tageszeituhr kann auch von der CPU-Konfigurationssoftware aus gelesen und eingestellt werden. Beachten Sie, dass Sie die Zeituhr mit dem Hand-Programmiergerät nicht verändern können, solange der Schlüsselschalterschutz aktiv ist.

Die Tageszeituhr ist so ausgelegt, dass sie die monatlichen und jährlichen Übergänge sowie die Schaltjahre bis zum Jahr 2079 automatisch bewältigen kann.

Überwachungszeitgeber

Eine Zeitüberwachung (Watchdog) in der SPS Series 90-30 soll Fehlerzustände erkennen, die zu einer außergewöhnlich langen Zyklusdauer führen. Die Überwachungsfunktion ist fest auf 200 Millisekunden (500 Millisekunden bei CPU 351/352) eingestellt, dieser Wert kann nicht verändert werden. Jeweils zu Beginn eines Zyklus läuft die Zeitüberwachung wieder bei Null an.

Wird bei einer CPU 331 oder einem kleineren Modell der CPUs 90-30 der Wert der Zeitüberwachung überschritten, erlischt die OK LED, die CPU wird rückgesetzt und vollständig abgeschaltet und die Ausgänge nehmen ihre voreingestellten Zustände ein. Datenaustausch ist nicht mehr möglich und alle Mikroprozessoren auf allen Modulen sind angehalten. Dieser Zustand kann nur behoben werden, indem Sie die Versorgungsspannung des CPU-Chassis aus- und wieder einschalten. Bei der 90-20, der Series 90 Micro und bei CPUs 90-30 ab Modell 340 aufwärts wird beim Ansprechen der Zeitüberwachung die CPU rückgesetzt. Die CPU führt dann die Einschaltroutine durch, erzeugt einen Zeitüberwachungsfehler und geht in **STOP**-Modus.

Verstrichene Stromausfallzeit

Mit dem Zeitgeber für die verstrichene Stromausfallzeit wird festgestellt, wie lange die SPS abgeschaltet war. Beim Abschalten der SPS wird der Zeitgeber auf 0 gesetzt und beginnt mit der Zeitzählung. Beim Einschalten der SPS wird die Zeitzählung gestoppt und der Wert bleibt erhalten. Mit der in Kapitel 12 beschriebenen Dienstanforderung #29 kann der Wert dieses Zeitgebers ausgelesen werden.

Hinweis

Diese Funktion steht nur in Series 90-30 CPUs ab Modell 331 zur Verfügung.

Zeitgeber für konstante Zyklusdauer

Der Zeitgeber für konstante Zyklusdauer steuert die Länge eines Programmzyklus, wenn die SPS Series 90-30 im Modus **KONSTANTE ZYKLUSDAUER** arbeitet. In dieser Betriebsart benötigt jeder Zyklus die gleiche Zeit. Im allgemeinen ist bei den meisten Anwenderprogrammen die in den einzelnen Zyklen benötige Zeit für Eingabezyklus, Programmbearbeitungszyklus und Ausgabezyklus nie genau gleich. Der Wert des Zeitgebers für konstante Zyklusdauer wird über das Programmiergerät im Bereich zwischen 5 Millisekunden und dem Wert der Zeitüberwachung (Watchdog) eingestellt. Der Standardwert ist 100 Millisekunden.

Läuft der Zeitgeber für konstante Zyklusdauer ab, ehe der Zyklus beendet ist, und wurde beim vorhergehenden Zyklus die Zeit nicht überschritten, trägt die SPS einen Zykluszeitüberschreitungsalarm in die SPS-Fehlertabelle ein. Zu Beginn des nächsten Zyklus setzt die SPS den Fehlerkontakt OV_SWP. Dieser Kontakt wird wieder rückgesetzt, wenn die SPS nicht im Modus **KONSTANTE ZYKLUSDAUER** ist oder wenn beim vorherigen Zyklus die Zeit nicht überschritten wurde.

Zeitkontakte

Die SPS Series 90 besitzt vier Zeitkontakte mit Einschaltwerten von 0,01 s, 0,1 s, 1,0 s und 1 min. Der Zustand dieser Kontakte ändert sich nicht, während der Zyklusausführung. Die von den Kontakten gelieferten Impulse haben ein 1:1-Taktverhältnis. Die Kontakte werden als T_10MS (0,01 s), T_100MS (0,1 s), T_SEC (1,0 s) und T_MIN (1 min) angesprochen.

Das nachstehende Taktdiagramm zeigt das Schaltspiel dieser Kontakte.

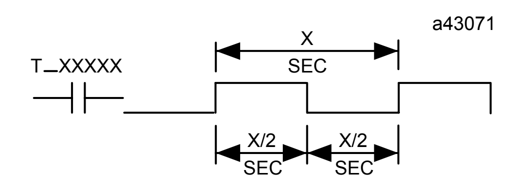


Abbildung 2-6. Taktdiagramm der Zeitkontakte

Abschnitt 5: Systemsicherheit

Das Sicherheitskonzept der SPS Series 90-20, 90-30 ist so ausgelegt, dass unberechtigte Änderungen des SPS-Inhalts verhindert werden können. Die SPS kennt vier Sicherheitsebenen. Die erste Ebene, die immer zugänglich ist, gestattet nur das Lesen der SPS-Daten, das Anwenderprogramm kann hier nicht verändert werden. Der Zugriff zu den anderen drei Ebenen wird durch Passworte geschützt.

Je höher die Ebene, desto größer die Anzahl der Privilegien. Zusätzlich zu den auf einer Ebene gewährten Privilegien sind die Privilegien aller darunterliegenden Ebenen verfügbar. Die Ebenen und ihre Privilegien sind:

Privileg-ebene	Beschreibung
Ebene 1	Mit Ausnahme der Passworte können alle Daten gelesen werden. Hierzu gehören alle Datenspeicher (%I, %Q, %AQ, %R, usw.), Fehlertabellen und alle Programmblocktypen (Daten, Werte und Konstanten). Es dürfen keine Werte in der SPS verändert werden.
Ebene 2	Auf dieser Ebene ist schreibender Zugriff auf die Datenspeicher (%I, %R, usw.) möglich.
Ebene 3	Auf dieser Ebene ist Schreibzugriff auf das Anwenderprogramm in **STOP**-Modus möglich.
Ebene 4	Dies ist die Standardebene für Systeme, bei denen keine Passworte eingestellt sind. Die Standardebene eines Systems mit Passworten ist die jeweils höchste ungeschützte Ebene. In dieser höchsten Ebene besteht Lese- und Schreibzugriff zu allen Speichern und Passworten im **RUN**- und im **STOP**-Modus (Konfigurationsdaten können im **RUN**-Modus nicht verändert werden).

Passworte

Für jede Privilegebene in der SPS gibt es ein Passwort. Für den Zugriff auf Ebene 1 kann kein Passwort eingestellt werden. Sie können für jede Ebene ein eigenes Passwort verwenden oder das gleiche Passwort für mehrere Ebenen vereinbaren. Ein Passwort besteht aus 1 bis 4 ASCII-Zeichen, es kann nur über die Programmiersoftware oder das Hand-Programmiergerät eingegeben oder geändert werden.

Ein Wechsel einer Privilegebene ist nur solange wirksam, wie der Datenverkehr zwischen SPS und Programmiergerät intakt ist. Es sind keine Aktivitäten erforderlich, die Verbindung darf jedoch nicht unterbrochen werden. Werden über einen Zeitraum von 15 Minuten keine Daten ausgetauscht, kehrt das System zur höchsten ungeschützten Ebene zurück.

Wird die Verbindung zur SPS hergestellt, fragt die Programmiersoftware die Schutzzustände der einzelnen Privilegebenen von der SPS ab. Anschließend fordert die Programmiersoftware die SPS auf, auf die höchste ungeschützte Ebene zu gehen, wobei die Programmiersoftware Zugang zur höchsten ungeschützten Ebene erhält, ohne eine bestimmte Ebene anfordern zu müssen. Wird das Hand-Programmiergerät an die SPS angeschlossen, geht die SPS wieder auf die höchste ungeschützte Ebene zurück.

Wechsel der Privilegebene

Ein Programmiergerät fordert einen Wechsel der Privilegebene, indem es die neue Privilegebene und das Passwort für diese Ebene liefert. Der Wechsel der Privilegebene wird abgelehnt, wenn das vom Programmiergerät gesendete Passwort nicht mit dem in der SPS-Passwortzugriffstabelle gespeicherten Passwort für die angeforderte Ebene übereinstimmt. In diesem Fall wird die aktuelle Privilegebene beibehalten und es findet kein Wechsel statt. Bei einem Versuch, mit dem Hand-Programmiergerät auf Daten in der SPS zuzugreifen oder diese zu verändern, ohne dass die richtige Privilegebene eingestellt wurde, reagiert das Hand-Programmiergerät mit einer Fehlermeldung und der Zugriff wird verweigert.

Sperren/freigeben von Unterprogrammen

Unterprogrammblöcke können mit der Block-Verriegelungsfunktion der Programmiersoftware gesperrt oder freigegeben werden. Es gibt zwei Sperrarten:

Sperre	Beschreibung
Anzeige	Bei Sperre können Sie nicht in dieses Unterprogramm zoomen.
Editieren	Bei Sperre können Sie die Daten im Unterprogramm nicht editieren.

Sofern ein Unterprogramm nicht permanent für Anzeigen oder Editieren gesperrt ist, können Sie die Sperren im Blockdeklarationseditor aufheben.

Auf ein für das Anschauen gesperrtes Unterprogramm kann eine Such- oder Suchen/Ersetzen-Funktion angewandt werden. Wird das gesuchte Objekt in einem solchen Unterprogramm gefunden, erscheint anstelle der Logik eine der folgenden Meldungen auf dem Bildschirm:

Gefunden in gesperrtem Block <block_name> (Weiter/Ende)

oder

Schreiben nicht möglich auf gesperrten Block <block_name> (Weiter/Ende)

Sie können den Suchvorgang nun fortsetzen oder abbrechen.

Ordner, die gesperrte Unterprogramme enthalten, können gelöscht werden. Werden die Ordnerfunktionen Kopieren, Backup und Restore der Programmiersoftware auf einen Ordner mit gesperrten Unterprogrammen angewandt, dann bleiben diese Sperren erhalten.

Permanentes Sperren eines Unterprogramms

Zusätzlich zur Anzeige- (VIEW LOCK) und Editiersperre (EDIT LOCK) gibt es noch zwei Arten von permanenten Sperren. Wurde für ein Unterprogramm eine permanente Anzeigesperre (PERMANENT VIEW LOCK) gesetzt, dann können Sie in dieses Unterprogramm nicht mehr hinein zoomen. Wurde eine permanente Editiersperre (PERMANENT EDIT LOCK) gesetzt, dann können Sie das betreffende Unterprogramm nicht mehr editieren.

Achtung

Im Gegensatz zu den normalen Sperren VIEW LOCK und EDIT LOCK können gesetzte permanente Sperren nicht mehr aufgehoben werden.

Eine permanente Editiersperre (PERMANENT EDIT LOCK) kann nur noch in eine permanente Anzeigesperre (PERMANENT VIEW LOCK) umgewandelt werden. Eine permanente Anzeigesperre (PERMANENT VIEW LOCK) kann in keine andere Sperre umgewandelt werden.

Abschnitt 6: E/A-System der SPS Series 90-30, 90-20 und Micro,

Die E/A-Systeme der SPS bilden die Schnittstelle zwischen der SPS Series 90-30 und den anwenderseitigen Geräten. Die E/A der Series 90-30 wird Series 90-30 E/A genannt. Die Module der Series 90-30 E/A werden direkt in die Steckplätze des CPU-Chassis und/oder - bei der SPS 90-30 ab Modell 331 aufwärts - in die Steckplätze der Erweiterungschassis gesteckt. Bei den E/A-Systemen der Modelle 331, 340 und 341 können bis zu 49 Series 90-30 E/A-Module eingebaut werden (5 Chassis). Bei den E/A-Systemen Modell 351 und 352 sind bis zu 79 Series 90-30 E/A-Module möglich (8 Chassis). Ein Chassis der Modelle 311 oder 313 der SPS Series 90-30 mit 5 Steckplätzen kann bis zu 5 Series 90-30 E/A-Module aufnehmen, ein Chassis mit 10 Steckplätzen bis zu 10 Module.

Die nachstehende Abbildung zeigt die E/A-Struktur der SPS Series 90-30.

SPS-E/A-System

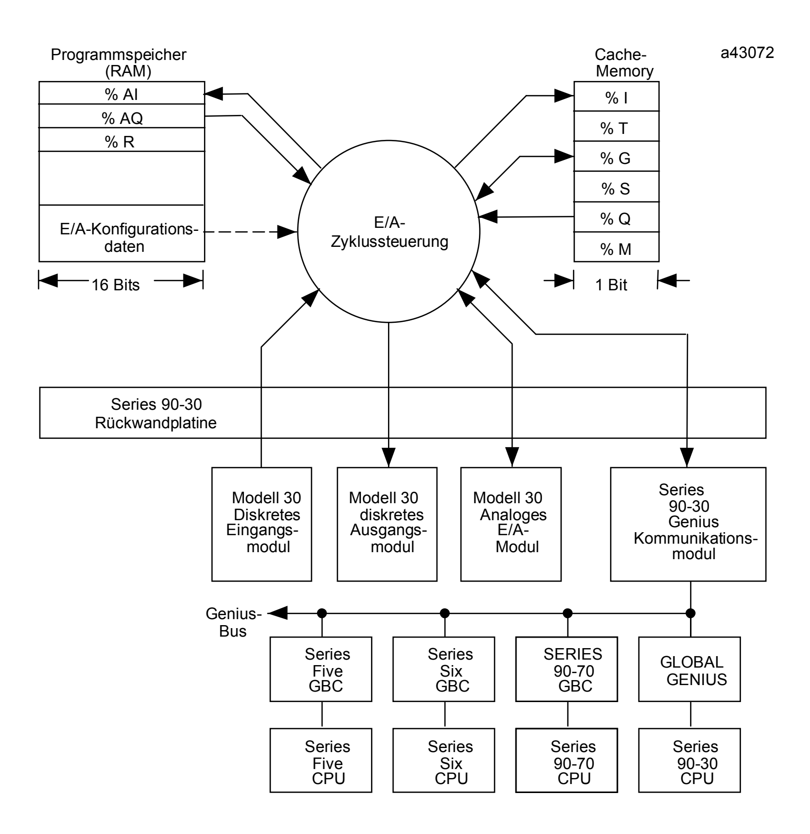


Abbildung 2-7. E/A-Struktur der Series 90-30

Hinweis

Die Struktur in der vorstehenden Abbildung ist spezifisch für die E/A der 90-30. Intelligente Module und Zusatzmodule sind nicht am E/A-Zyklus beteiligt, sie benutzen das Systemkommunikationsfenster. Informationen über die E/A-Struktur der 90-20 finden Sie in GFK-0551, *SPS Series 90™-20, Anwenderhandbuch*. Informationen über die E/A-Struktur der Micro-SPS finden Sie in GFK-1065, *SPS Series 90™ Micro, Anwenderhandbuch*.

Series 90-30 E/A-Module

Es gibt fünf verschiedene Typen der Series 90-30 E/A-Module: Diskrete Eingangsmodule, diskrete Ausgangsmodule, analoge Eingangsmodule, analoge Ausgangsmodule und Zusatzmodule. In der nachstehenden Tabelle finden Sie eine Auflistung der Series 90-30 E/A-Module mit Bestellnummer, Anzahl E/A-Punkte und einer kurzen Beschreibung.

Hinweis

Es ist möglich, dass einige der in der Tabelle enthaltenen E/A-Module zum Zeitpunkt der Drucklegung dieses Handbuches noch nicht verfügbar waren. Ihr GE Fanuc Distributor kann Ihnen genaue Auskünfte über die jeweilige Liefersituation geben. Die technischen Daten und Anschlusspläne der einzelnen Series 90-30 E/A-Module finden Sie in GFK-0898, *Series 90-30 E/A-Module, Technische Daten, Handbuch*.

Abbildung 2-8. Series 90-30 E/A-Module

Bestell-nummer	Punkte	Beschreibung	Hand-buch
		Diskrete Eingangsmodule	
IC693MDL230	8	120 VAC, potentialgetrennt	GFK-0898
IC693MDL231	8	240 VAC, potentialgetrennt	GFK-0898
IC693MDL240	16	120 VAC	GFK-0898
IC693MDL241	16	24 VAC/DC pos./neg. Logik	GFK-0898
IC693MDL630	8	24 VDC pos. Logik	GFK-0898
IC693MDL632	8	125 VDC pos./neg. Logik	GFK-0898
IC693MDL633	8	24 VDC neg. Logik	GFK-0898
IC693MDL634	8	24 VDC pos./neg. Logik	GFK-0898
IC693MDL640	16	24 VDC pos. Logik	GFK-0898
IC693MDL641	16	24 VDC neg. Logik	GFK-0898
IC693MDL643	16	24 VDC pos. Logik	GFK-0898
IC693MDL644	16	24 VDC neg. Logik	GFK-0898
IC693MDL645	16	24 VDC pos./neg. Logik	GFK-0898
IC693MDL646	16	24 VDC pos./neg. Logik, schnell	GFK-0898
IC693MDL652	32	24 VDC pos./neg. Logik	GFK-0898
IC693MDL653	32	24 VDC pos./neg. Logik, schnell	GFK-0898
IC693MDL654	32	5/12 VDC (TTL) pos./neg. Logik	GFK-0898
IC693MDL655	32	24 VDC pos./neg. Logik	GFK-0898
IC693ACC300	8/16	Eingangssimulator	GFK-0898

Tabelle 2-8. Series 90-30 E/A-Module - Fortsetzung

Bestell-nummer	Punkte	Beschreibung	Hand-buch
		Diskrete Ausgangsmodule	
IC693MDL310	12	120 VAC, 0,5 A	GFK-0898
IC693MDL330	8	120/240 VAC, 2 A	GFK-0898
IC693MDL340	16	120 VAC, 0,5 A	GFK-0898
IC693MDL390	5	120/240 VAC potentialgetrennt, 2 A	GFK-0898
IC693MDL730	8	12/24 VDC positive Logik, 2 A	GFK-0898
IC693MDL731	8	12/24 VDC negative Logik, 2 A	GFK-0898
IC693MDL732	8	12/24 VDC positive Logik, 0.5 A	GFK-0898
IC693MDL733	8	12/24 VDC negative Logik, 0.5 A	GFK-0898
IC693MDL734	6	125 VDC pos./neg. Logik, 2 A	GFK-0898
IC693MDL740	16	12/24 VDC positive Logik, 0.5 A	GFK-0898
IC693MDL741	16	12/24 VDC negative Logik, 0.5 A	GFK-0898
IC693MDL742	16	12/24 VDC positive Logik, 1 A	GFK-0898
IC693MDL750	32	12/24 VDC negative Logik	GFK-0898
IC693MDL751	32	12/24 VDC positive Logik, 0.3 A	GFK-0898
IC693MDL752	32	5/24 VDC (TTL) negative Logik, 0,5 A	GFK-0898
IC693MDL753	32	12/24 VDC pos./neg. Logik, 0,5 A	GFK-0898
IC693MDL930	8	Relais, Schliesser, 4 A potentialgetrennt	GFK-0898
IC693MDL931	8	Relais, BC, potentialgetrennt	GFK-0898
IC693MDL940	16	Relais, Schliesser, 2 A	GFK-0898
		Ein-/Ausgangsmodule	
IC693MDR390	8/8	24 VDC Eingang, Relaisausgang	GFK-0898
IC693MAR590	8/8	120 VAC Eingang, Relaisausgang	GFK-0898
		Analogmodule	
IC693ALG220	4-Kanal	Analogeingang, Spannung	GFK-0898
IC693ALG221	4-Kanal	Analogeingang, Strom	GFK-0898
IC693ALG222	16	Analogeingang, Spannung	GFK-0898
IC693ALG223	16	Analogeingang, Strom	GFK-0898
IC693ALG390	2-Kanal	Analogausgang, Spannung	GFK-0898
IC693ALG391	2-Kanal	Analogausgang, Strom	GFK-0898
IC693ALG392	8-Kanal	Analogausgang, Strom/Spannung	GFK-0898
IC693ALG442	4/2	Kombi-Analogmodul, Strom/Spannung, Ein/Aus	GFK-0898

Tabelle 2-8. Series 90-30 E/A-Module - Fortsetzung

Bestell-nummer	Beschreibung	Hand-buch
	Zusatzmodule	
IC693APU300	Schneller Zähler	GFK-0293
IC693APU301	Power Mate APM-Modul, 1 Achse–Nachlaufmodus	GFK-0781
IC693APU301	Power Mate APM-Modul, 1 Achse–Standardmodus	GFK-0840
IC693APU302	Power Mate APM-Modul, 2 Achsen–Nachlaufmodus	GFK-0781
IC693APU302	Power Mate APM-Modul, 2 Achsen–Standardmodus	GFK-0840
IC693MCS001/2	Power Mate J Motion Control System (1 und 2 Achse(n))	GFK-1256
IC693APU305	E/A-Prozessormodule	GFK-1028
IC693CMM321	Ethernet-Kommunikationsmodul	GFK-1084
IC693ADC311	Alphanumerischer Anzeige-Coprozessor	GFK-0521
IC693BEM331	Genius-Buscontroller	GFK-1034
IC693BEM320	Schnittstellenmodul zu CNC und SPS Series 90-70 (Slave)	GFK-0631
IC693BEM321	Schnittstellenmodul zu CNC und SPS Series 90-70 (Master)	GFK-0823
IC693CMM311	Kommunikations-Coprozessormodul	GFK-0582
IC693CMM301	Genius-Kommunikationsmodul	GFK-0412
IC693CMM302	Erweitertes Genius-Kommunikationsmodul	GFK-0695
IC693PCM300	PCM, 160 kByte (35 kByte MegaBasic-Anwenderprogramm)	GFK-0255
IC693PCM301	PCM, 192 kByte (47 kByte MegaBasic-Anwenderprogramm)	GFK-0255
IC693PCM311	PCM, 640 kByte (190 kByte MegaBasic-Anwenderprog.)	GFK-0255

E/A-Datenformate

Diskrete Ein- und Ausgabedaten werden als Bits im Cache-Memory (Statustabelle) gespeichert. Analoge Ein- und Ausgangsdaten werden als Worte gespeichert und liegen speicherresident in einem für diesen Zweck reservierten Teil des Anwender-RAMs.

Standardbedingungen für die Series 90-30 Ausgangsmodule

In der Standardeinstellung gehen die Ausgänge der Series 90-30 Ausgangsmodule beim Einschalten der Versorgungsspannung auf AUS. Sie halten diesen Wert bis zum ersten Ausgabezyklus von der SPS. Analog-Ausgangsmodule können über einen Brückenstecker (neben dem abnehmbaren Klemmenblock des Moduls) so eingestellt werden, dass sie standardmäßig auf Null gehen oder ihren letzten Zustand halten. Da Analog-Ausgangsmodule auch über eine externe Stromquelle versorgt werden können, können sie selbst bei stromloser SPS ihren Betrieb im eingestellten Standardmodus fortsetzen.

Diagnosedaten

Diagnosebits im %S-Speicher zeigen Ausfälle von E/A-Modulen und Diskrepanzen bei der E/A-Konfiguration an. Es gibt keine Diagnosedaten für einzelnen E/A-Punkte. Weitere Informationen zur Fehlerbearbeitung finden Sie in Kapitel 3 "Fehlerbeschreibungen und Fehlerbehebung" in diesem Handbuch.

Global Data

Genius Global Data

Mit Genius Global Data können bei der SPS Series 90-30 mit hoher Geschwindigkeit Daten von mehreren CPUs gemeinsam benutzt werden. Mit dem Genius-Buscontroller (IC693BEM331) in der CPU (ab Version 5) und dem erweiterten Genius-Kommunikationsmodul (IC693CMM302) können bis zu 128 Datenbyte zu anderen SPS oder Computern gesendet werden. Von jedem der bis zu 30 anderen Teilnehmern am Netz können bis zu 128 Byte empfangen werden. Zum Senden und Empfangen der Daten kann jeder Speichertyp verwendet werden, nicht nur %G-Globalbits. Das ursprüngliche Genius-Kommunikationsmodul (IC693CMM301) war auf feste %G-Adressen beschränkt und konnte pro serieller Busadresse von SBA 16 bis 23 nur 32 Bits austauschen. Dieses erste Modul sollte nicht mehr verwendet werden, da das erweiterte GCM mehr als die 100-fache Kapazität besitzt.

Global Data können gemeinsam benutzt werden von speicherprogrammierbaren Steuerungen der Series Five, Series Six und Series 90, die am gleichen Genius-E/A-Bus angeschlossen sind.

Ethernet Datenverkehr

Die CPU Modell 364 (Ausgabestand 9.0 und höher) unterstützt den Anschluss an ein Ethernet-Netzwerk über einen (jedoch nicht beide) der beiden integrierten Ethernet-Ports. Es stehen AAUI- und 10BaseT-Ports zur Verfügung. Die CPU Modell 364 (Ausgabestand 9.10 oder höher) ist die einzige CPU der Series 90-30, die EGD unterstützt.

Die CPU Modell 364 unterstützt Ethernet Global Data (EGD). EGD ist insofern ähnlich zu Genius Global Data, dass es einem Gerät (dem Sender) die Übertragung von Daten zu einem oder mehreren anderen Geräten (den Empfängern) im Netzwerk gestattet. EGD wird von der Logicmaster 90 Software nicht unterstützt (es wird das Programmierpaket für SPS Series 90 auf der Grundlage von Windows benötigt).

Modell 20 E/A-Module

Es sind folgende E/A-Module für die SPS Series 90-20 lieferbar. Die einzelnen Module sind mit ihrer Bestellnummer, der Anzahl E/A-Punkte sowie einer kurzen Beschreibung aufgelistet. Die E/A ist zusammen mit der Stromversorgung in einem Chassis integriert. Die technischen Daten und Anschlusspläne der einzelnen Modell 20 E/A-Module finden Sie in GFK-0551, *SPS Series 90-20,* Anwenderhandbuch,.

Bestell-nummer	Beschreibung	E/A-Punkte
IC692MAA541	E/A- und Stromversorgungs-Grundmodul, 120 VAC Ein/120 VAC Aus/120 VAC Stromversorgung	16 Ein/12 Aus
IC692MDR541	E/A- und Stromversorgungs-Grundmodul 24 VDC Ein/Relais Aus/120 VAC Stromversorgung	16 Ein/12 Aus
IC692MDR741	E/A- und Stromversorgungs-Grundmodul 24 VDC Ein/Relais Aus/240 VAC Stromversorgung	16 Ein/12 Aus
IC692CPU211	CPU-Modul, Modell CPU 211	Nicht zutreffend

Konfiguration und Programmierung

Bei der Konfiguration werden den Hardwaremodulen im System logische Adressen sowie weitere Eigenschaften zugeordnet. Die Konfiguration kann mit der Konfigurationssoftware oder dem Hand-Programmiergerät vor oder nach der Programmierung durchgeführt werden. Es wird jedoch empfohlen, die Konfiguration zuerst durchzuführen. In GFK-0466, *Programmiersoftware, Anwenderhandbuch* finden Sie Entscheidungshilfen dazu, ob Sie jetzt mit der Programmierung anfangen sollen.

Bei der Programmierung wird ein Anwenderprogramm für eine SPS erstellt. Da die SPS der Series 90-30, 90-20 und Series 90 Micro einen gemeinsamen Befehlssatz besitzen, können alle drei Systeme mit der Logicmaster 90-30 Software programmiert werden. In den Kapiteln 4 bis 12 werden die Programmieranweisungen beschrieben, mit denen die Kontaktplanprogramme für die speicherprogrammierbaren Steuerungen der Series 90-30 und Series 90-20 erstellt werden können.

Haben Sie die Logicmaster 90-30/20/Micro Programmiersoftware noch nicht installiert, finden Sie entsprechende Anweisungen in GFK-0466, *Programmiersoftware, Anwenderhandbuch*. Das Anwenderhandbuch erläutert, wie Programme erstellt, übertragen, bearbeitet und ausgedruckt werden können.

Kapitel 3 Erläuterung und Behebung von Fehlern

Dieses Kapitel bietet Hilfestellung bei der Fehlersuche in SPS-System der Series 90-30, 90-20 und Micro. Es erläutert die Fehlerbeschreibungen in der SPS-Fehlertabelle und die Fehlerkategorien in der E/A-Fehlertabelle.

In diesem Kapitel werden bei der Erläuterung der einzelnen Fehler die Fehlerbeschreibung für die SPS-Fehlertabelle bzw. die Fehlerkategorie für die E/A-Fehlertabelle aufgelistet. Suchen Sie die Fehlerbeschreibung oder Fehlerkategorie, die dem Eintrag der entsprechenden Fehlertabelle in Ihrem Programmiergerätemenü entspricht. Darunter finden Sie dann eine Beschreibung der Fehlerursache und eine Anleitung, wie Sie den Fehler beheben können.

Kapitel 3 enthält folgende Abschnitte:

Abschnitt 1: Fehlerbehandlung

Hinweis

Diese Informationen zur Fehlerbehandlung beziehen sich auf Systeme, die mit der Logicmaster 90-30/20/Micro Software programmiert wurden.

In einem SPS-System Series 90-30, 90-20 oder Series 90 Micro spricht man von einem Fehler, wenn bestimmten Ausfälle oder Zustände auftreten, die Betrieb und Leistung des Systems beeinträchtigen. Diese Zustände, wie zum Beispiel der Verlust eines E/A-Moduls oder Chassis, können die Fähigkeit der SPS zur Steuerung einer Maschine oder eines Prozesses beeinträchtigen. Diese Zustände können aber auch Vorteile bringen, wenn zum Beispiel ein neues Modul online geschaltet wird und dann benutzt werden kann. In anderen Fällen dienen diese Zustände nur als eine Warnung, wie bei einem Batterieausfallsignal, das anzeigt, dass die für den Speicherschutz eingesetzte Batterie ausgewechselt werden muss.

Alarmprozessor

Der Zustand oder Ausfall selbst wird Fehler genannt. Empfängt und bearbeitet die CPU einen Fehler, sprechen wir von einem Alarm. Die Software in der CPU, die diese Zustände bearbeitet, wird Alarmprozessor genannt. Die Programmiersoftware bildet für den Alarmprozessor die Schnittstelle zum Anwender. Jeder erkannte Fehler wird in eine Fehlertabelle eingetragen und je nachdem entweder im SPS-Fehlertabellenmenü oder im E/A-Fehlertabellenmenü angezeigt.

Fehlerklassen

Die SPS Series 90-30, 90-20 und Micro können mehrere Fehlerklassen erkennen. Hierzu gehören interne Fehler, externe Fehler und Betriebsfehler.

Fehlerklasse	Beispiele
Interne Fehler	Nicht reagierende Module Niedrige Batteriespannung Prüfsummenfehler im Speicher
Externe E/A-Fehler	Verlust von Chassis oder Modul Hinzufügen von Chassis oder Modul
Betriebsfehler	Kommunikationsfehler Konfigurationsfehler Passwortfehler

Hinweis

Spezielle Informationen zur Fehlerbehandlung bei der Micro SPS finden Sie in GFK-1065, *SPS Series 90 Micro, Anwenderhandbuch*.

Systemreaktion auf Fehler

Ein Hardwarefehler wird entweder toleriert oder macht eine Abschaltung des Systems erforderlich. Ein E/A-Fehler kann zwar vom SPS-System toleriert werden, aber intolerabel für die Anwendung oder den gesteuerten Prozess sein. Betriebsfehler werden normalerweise toleriert. Bei den SPS Series 90-30, 90-20 und Micro besitzen die Fehler zwei Attribute:

Attribut	Beschreibung
Betroffene Fehlertabelle	E/A-Fehlertabelle SPS-Fehlertabelle
Fehlerwirkung	Fatal Diagnose Informatorisch

Fehlertabellen

In der SPS werden zwei Fehlertabellen unterhalten: Während in die E/A-Fehlertabelle Fehler eingetragen werden, die sich auf das E/A-System beziehen, werden in der SPS-Fehlertabelle alle anderen Fehler protokolliert. In der nachstehenden Tabelle sind die Fehlergruppen, ihre Fehlerwirkung, die betroffenen Fehlertabellen und die symbolischen Adressen der betroffenen diskreten %S-Systempunkte aufgelistet.

Tabelle 3-1. Zusammenfassung der Fehler

Fehlergruppe	Fehlerwirkung	Fehlertabelle	Diskrete Spezial-Fehlerreferenzen			
Verlorenes oder fehlendes E/A-Modul	Diagnose	E/A	io_flt	any_flt	io_pres	los_iom
Verlorenes oder fehlendes Zusatzmodul	Diagnose	SPS	sy_flt	any_flt	sy_pres	los_sio
Diskrepanz bei Systemkonfiguration	Fatal	SPS	sy_flt	any_flt	sy_pres	cfg_mm
SPS CPU Hardwarefehler	Fatal	SPS	sy_flt	any_flt	sy_pres	hrd_cpu
Programm-Prüfsummenfehler	Fatal	SPS	sy_flt	any_flt	sy_pres	PB_SUM
Niedrige Batteriespannung	Diagnose	SPS	sy_flt	any_flt	sy_pres	LOW_BAT
SPS-Fehlertabelle ist voll	Diagnose	—	SY_FULL			
E/A-Fehlertabelle ist voll	Diagnose	—	IO_FULL			
Anwendungsfehler	Diagnose	SPS	sy_flt	any_flt	sy_pres	APL_FLT
Anwenderprogramm fehlt	Informatorisch	SPS	sy_flt	any_flt	sy_pres	no_prog
Fehler im Anwender-RAM	Fatal	SPS	sy_flt	any_flt	sy_pres	BAD_RAM
Passwortfehler	Diagnose	SPS	sy_flt	any_flt	sy_pres	BAD_PWD
SPS-Softwarefehler	Fatal	SPS	sy_flt	any_flt	sy_pres	SFT_CPU
SPS-Speicherfehler	Fatal	SPS	sy_flt	any_flt	sy_pres	STOR_ER
Konstante Zykluszeit überschritten	Diagnose	SPS	sy_flt	any_flt	sy_pres	OV_SWP
Unbekannter SPS-Fehler	Fatal	SPS	sy_flt	any_flt	sy_pres	
Unbekannter E/A-Fehler	Fatal	E/A	io_flt	any_flt	io_pres	

Fehlerwirkung

Ein Fehler kann fatal, diagnostisch oder informatorisch sein.

Ein fataler Fehler wird in die entsprechende Tabelle eingetragen, setzt Diagnosevariablen und hält das System an. Ein Diagnosefehler wird in die entsprechende Tabelle eingetragen und setzt Diagnosevariablen. Ein informatorischer Fehler wird nur in die entsprechende Tabelle eingetragen.

Die nachstehende Tabelle zeigt mögliche Fehlerwirkungen.

Tabelle 3-2. Fehlerwirkungen

Fehlerwirkung	Reaktion der CPU
Fatal	Fehlereintrag in Fehlertabelle Fehlerreferenzen setzen Auf **STOP** gehen
Diagnose	Fehlereintrag in Fehlertabelle Fehlerreferenzen setzen
Informatorisch	Fehlereintrag in Fehlertabelle

Wird ein Fehler erkannt, veranlasst die CPU die entsprechende Fehlerwirkung. Die Fehlerwirkungen in der SPS Series 90-30, Series 90-20 und Series 90 Micro sind nicht konfigurierbar.

Fehlerreferenzen

Bei den SPS Series 90-30, 90-20 und Micro gibt es nur einen Typ Fehlerreferenzen, die summarischen Fehlerreferenzen. Summarische Fehlerreferenzen werden gesetzt um anzugeben, welcher Fehler aufgetreten ist. Die Fehlerreferenz bleibt solange gesetzt, bis sie vom Anwenderprogramm gelöscht wird oder bis die SPS gelöscht wird.

Das nachfolgende Beispiel zeigt, wie ein Fehlerbit gesetzt und dann wieder gelöscht wird. In dem Beispiel wird die Spule licht_01 durchgeschaltet, wenn die Zykluszeit überschritten wird. Das Licht und der Kontakt OV_SWP bleiben solange eingeschaltet, bis der Kontakt %I0359 geschlossen wird.

```
| ov_swp                                                        licht_01
|——] [——————————————————————————————————————————————————————————( )—
|
|%I0359                                                           ov_swp
|——] [——————————————————————————————————————————————————————————(R)—
|
```

Definitionen der Fehlerreferenz

Der Alarmprozessor verwaltet die Zustände von 128 Systembits im %S-Speicher. Mit diesen Fehlerreferenzen kann angezeigt werden, wo ein Fehler aufgetreten ist und von welchem Typ dieser Fehler ist. Fehlerreferenzen sind den %S-, %SA-, %SB- und %SC-Speichern zugeordnet, jede Fehlerreferenz besitzt eine symbolische Adresse. Die Referenzen können bei Bedarf im Anwenderprogramm verwendet werden. Eine Liste der Systemzustandsreferenzen finden Sie in Kapitel 2 "Systembetrieb" dieses Handbuchs.

Weitere Fehlerauswirkungen

Zwei der zuvor beschriebenen Fehler haben noch Nebenwirkungen. Diese werden in der nachstehenden Tabelle beschrieben.

Nebenwirkung	Beschreibung
SPS CPU Softwarefehler	Wird ein SPS CPU Softwarefehler protokolliert, geht die CPU der Serie 90-30 bzw. 90-20 unmittelbar in einen speziellen **FEHLERZYLUS**-Modus. In diesem Modus ist keine Aktivität erlaubt. Zum Lösen dieses Zustands müssen Sie die SPS rücksetzen (d.h. Versorgungsspannung aus- und wieder einschalten).
Fehler beim Speichern von SPS-Folgen	Wird während einer Sequenzspeicherung (Speichern von Programmblöcken und anderen Daten, denen der spezielle Sequenzbeginn-Befehl vorangestellt ist und die mit dem Sequenzende-Befehl enden) die Datenverbindung zu dem die Speicherung durchführenden Programmiergerät unterbrochen oder tritt ein anderer Fehler auf, der das Laden beendet, wird der "Fehler beim Speichern von SPS-Sequenzen" eingetragen. Solange dieser Fehler im System ansteht geht die SPS nicht in **RUN**-Modus.

Anzeige der SPS-Fehlertabelle

Im SPS-Fehlertabellenmenü werden SPS-Fehler wie Passwortfehler, Diskrepanzen bei SPS-Konfiguration, Paritätsfehler und Kommunikationsfehler angezeigt.

Die Betriebsart der Programmiersoftware spielt dabei keine Rolle. Im **OFFLINE**-Modus werden keine Fehler angezeigt. Im **ONLINE**- oder **MONITOR**-Modus werden die SPS-Fehlerdaten angezeigt. Im **ONLINE**-Modus können Fehler gelöscht werden (diese Funktion kann durch ein Passwort geschützt werden).

Steht ein gelöschter Fehler weiterhin an, wird er nicht nochmals in die Tabelle eingetragen (Ausnahme: Fehler "Niedrige Batteriespannung").

Anzeige der E/A-Fehlertabelle

Im E/A-Fehlertabellenmenü werden E/A-Fehler wie zum Beispiel Schaltkreisfehler, Adressenkonflikte, fixierte Schaltkreise und Fehler am E/A-Bus angezeigt.

Die Betriebsart der Programmiersoftware spielt dabei keine Rolle. Im **OFFLINE**-Modus werden keine Fehler angezeigt. Im **ONLINE**- oder **MONITOR**-Modus werden die E/A-Fehlerdaten angezeigt. Im **ONLINE**-Modus können Fehler gelöscht werden (diese Funktion kann durch ein Passwort geschützt werden).

Steht ein gelöschter Fehler weiterhin an, wird er nicht nochmals in die Tabelle eingetragen.

Zugriff auf weitere Fehlerdaten

Die Fehlertabellen enthalten Grundinformationen zu dem Fehler. Weitere Angaben zu den einzelnen Fehlern erhalten Sie ebenso über die Programmiersoftware wie einen hexadezimalen Speicherabzug des Fehlers.

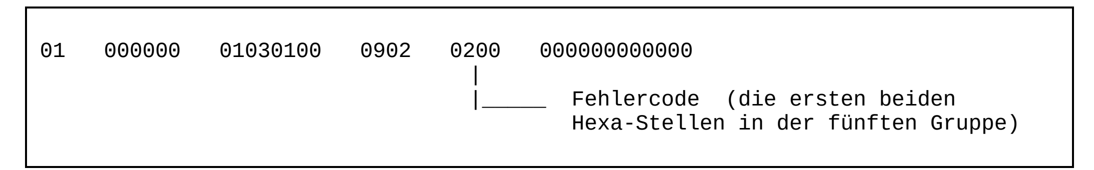
Im letzten Eintrag, Korrekturen, werden für die einzelnen Fehlerbeschreibungen in diesem Kapitel die Aktionen beschrieben, mit denen Sie den Fehler beheben können. Beachten Sie, dass bei einigen dieser Aktionen der folgende Hinweis steht:

Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.

Dieser Hinweis bedeutet, dass Sie der Serviceabteilung sagen müssen, was Sie direkt aus der Fehlertabelle ablesen können **und** welche hexadezimale Daten Sie sehen. Ihre Ansprechpartner in der Serviceabteilung können Ihnen dann weitere Anweisungen zur Fehlerbehebung geben.

Abschnitt 2: SPS-Fehlertabelle - Erläuterungen

Jede Fehlererläuterung enthält eine Fehlerbeschreibung und Anweisungen, wie der Fehler behoben werden kann. Zahlreiche Fehlerbeschreibungen haben mehrere Ursachen. In diesen Fällen wird der Fehlercode, der bei den zusätzlichen Fehlerinformationen angezeigt wird, zur Unterscheidung unterschiedlicher Fehlerzustände mit gleicher Fehlerbeschreibung benutzt. Der Fehlercode besteht aus den ersten beiden Hexadezimalstellen der fünften Zahlengruppe (siehe nachstehendes Beispiel).



```
01   000000   01030100   0902   0200   000000000000
                                  |
                                  |_____  Fehlercode  (die ersten beiden
                                          Hexa-Stellen in der fünften Gruppe)
```

Einige Fehler können auftreten, wenn der RAM auf der CPU-Platine der SPS fehlerbehaftet ist. Die gleichen Fehler können auch auftreten, wenn das System abgeschaltet wurde und die Batteriespannung zu niedrig war, um den Speicherinhalt zu erhalten. Um hier zu vermeiden, dass die gleichen Anweisungen häufig wiederholt werden müssen, wird bei möglichen Speicherfehlern als Fehlerursache lediglich angegeben:

Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.

Das heißt:

1. Wechseln Sie die Batterie aus, wenn das System abgeschaltet wurde. Die Batteriespannung kann für den Erhalt des Speicherinhalts zu niedrig sein.

2. Wechseln Sie die CPU-Platine der SPS aus. Die Schaltkreise dieser Platine können fehlerhaft sein.

Die nachstehende Tabelle soll Ihnen helfen, schnell eine bestimmte SPS-Fehlerbeschreibung in diesem Abschnitt zu finden. Die einzelnen Einträge sind in der Reihenfolge aufgeführt, wie sie auf dem Programmiergeräte-Bildschirm erscheinen.

Fehlerwirkungen

Bei einem **fatalen** Fehler geht die SPS am Ende des Zyklus, in dem der Fehler aufgetreten ist, auf **STOP**. **Diagnosefehler** werden protokolliert und es werden die entsprechenden Fehlerkontakte gesetzt. **Informatorische** Fehler werden nur in die SPS-Fehlertabelle eingetragen.

Verlorenes oder fehlendes Zusatzmodul

Die Fehlergruppe **Verlorenes oder fehlendes Zusatzmodul** tritt auf, wenn ein PCM, CMM oder ADC nicht antwortet. Der Fehler kann beim Einschalten auftreten, wenn das Modul fehlt, oder im Betrieb, wenn das Modul nicht antwortet. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

Fehlercode:	1, 42
Name:	Software-Rücksetzen von Zusatzmodul fehlgeschlagen
Beschreibung:	CPU der SPS kann nach Software-Rücksetzen den Datenverkehr mit dem Zusatzmodul nicht mehr aufnehmen.
Beseitigung:	(1) Versuchen Sie nochmals Software-Rücksetzen. (2) Wechseln Sie das Zusatzmodul aus. (3) Schalten Sie das System ab. Überprüfen Sie die Kabelanschlüsse und den Sitz des PCM im Chassis. (4) Wechseln Sie die Kabel aus.
Fehlercode:	Alle übrigen.
Name:	Modulfehler während Konfiguration.
Beschreibung:	Die SPS-Betriebssoftware erzeugt dieses Fehlermeldung, wenn ein Modul beim Einschalten oder beim Speichern der Konfiguration ausfällt.
Beseitigung:	(1) System abschalten. Wechseln Sie das Modul in diesem Chassis und Steckplatz aus.

Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul

Die Fehlergruppe **Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul** tritt auf, wenn ein Zusatzmodul (PCM, ADC, usw.) online geht, rückgesetzt wird oder im Chassis festgestellt wird, ohne dass es in der Konfiguration angegeben wurde. Die Fehlerwirkung bei dieser Gruppe ist **Diagnose**. Drei fehlerspezifische Byte enthalten zusätzliche Angaben zu dem Fehler.

Beseitigung:	(1) Aktualisieren Sie die Konfigurationsdatei und nehmen Sie das Modul mit auf. (2) Entfernen Sie das Modul aus dem System.

Diskrepanz bei Systemkonfiguration

Die Fehlergruppe **Diskrepanz bei Konfiguration** tritt auf, wenn das in einem Steckplatz eingebaute Modul nicht den Angaben in der Konfigurationsdatei entspricht. Die Fehlerwirkung ist **fatal**.

Fehlercode:	1
Name:	Diskrepanz bei Systemkonfiguration
Beschreibung:	Die SPS-Betriebssoftware (Systemkonfigurierer) erzeugt diesen Fehler, wenn das in einen Steckplatz eingebaute Modul nicht von dem Typ ist, der in der Konfigurationsdatei für diesen Steckplatz angegeben wurde, oder wenn das vorhandene Chassis nicht mit dem konfigurierten Chassistyp übereinstimmt.
Beseitigung:	Diskrepanz feststellen und Modul oder Chassis neu konfigurieren.
Fehlercode:	6
Name:	Diskrepanz bei Systemkonfiguration
Beschreibung:	Wie bei Fehlercode 1 tritt dieser Fehler auf, wenn das in einen Steckplatz eingebaute Modul nicht von dem Typ ist, der in der Konfigurationsdatei für diesen Steckplatz angegeben wurde, oder wenn das vorhandene Chassis nicht mit dem konfigurierten Chassistyp übereinstimmt.
Beseitigung:	Diskrepanz feststellen und Modul oder Chassis neu konfigurieren.
Fehlercode:	18
Name:	Nicht unterstützte Hardware
Beschreibung:	In einer 311, 313 oder 323 oder in einem Erweiterungschassis wurde ein PCM oder PCM-ähnliches Modul eingebaut.
Beseitigung:	PCM oder PCM-ähnliches Modul entfernen oder eine CPU einbauen, die das PCM unterstützt.
Fehlercode:	26
Name:	Modul belegt – Konfiguration von Modul noch nicht angenommen.
Beschreibung:	Das Modul kann derzeit keine neue Konfiguration annehmen, da es mit einem anderen Vorgang beschäftigt ist.
Beseitigung:	Lassen Sie das Modul den aktuellen Vorgang beenden und speichern Sie dann die Konfiguration nochmals.
Fehlercode:	51
Name:	END-Funktion aus SFC-Aktion heraus ausgeführt.
Beschreibung:	Die Lage einer END-Funktion in einem SFC-Programm oder in einem über SFC aufgerufenen Programm verursacht diesen Fehler.
Beseitigung:	END-Funktion aus SFC-Programm bzw. dem über SFC aufgerufenen Programm entfernen.

Softwarefehler bei Zusatzmodul

Die Fehlergruppe **Softwarefehler bei Zusatzmodul** tritt auf, wenn ein nicht behebbarer Softwarefehler in einem PCM- oder ADC-Modul auftritt. Die Fehlerwirkung dieser Gruppe ist **fatal**.

Fehlercode:	Alle
Name:	COMMREQ kommt zu oft
Beschreibung:	Die COMMREQ-Befehle werden schneller zu dem Modul geschickt, als sie dort bearbeitet werden können.
Beseitigung:	SPS-Programm ändern und COMMREQ-Häufigkeit reduzieren.

Programmblock-Prüfsummenfehler

Die Fehlergruppe **Programmblock-Prüfsummenfehler** tritt auf, wenn die CPU der SPS einen Fehler in den durch die SPS empfangenen Programmblöcken erkennt. Sie tritt auch auf, wenn die CPU der SPS einen Prüfsummenfehler im Einschalt-Speichertest oder während der Hintergrundprüfung im **RUN**-Modus erkennt. Die Fehlerwirkung dieser Gruppe ist **fatal**.

Fehlercode:	Alle
Name:	Programmblock-Prüfsummenfehler
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler bei einem verstümmelten Programmblock.
Beseitigung:	(1) SPS-Speicher löschen und Speicherversuch wiederholen. (2) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.

Signal für niedrige Batteriespannung

Die Fehlergruppe **Signal für niedrige Batteriespannung** tritt auf, wenn die CPU der SPS erkennt, dass die Spannung der Batterie in der SPS-Stromversorgung zu niedrig ist oder wenn ein Modul (z.B. das PCM) zu niedrige Batteriespannung meldet. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

Fehlercode:	0
Name:	Batterieausfallsignal
Beschreibung:	Die Batterie der CPU (oder eines anderen batteriegepufferten Moduls) ist leer.
Beseitigung:	Batterie auswechseln. Die Versorgungsspannung zum Chassis nicht ausschalten!
Fehlercode:	1
Name:	Signal "niedrige Batteriespannung"
Beschreibung:	Eine Batterie in der CPU oder einem anderen Modul meldet niedrige Spannung.
Beseitigung:	Batterie auswechseln. Die Versorgungsspannung zum Chassis nicht ausschalten!

Konstante Zyklusdauer überschritten

Die Fehlergruppe **Konstante Zyklusdauer überschritten** tritt auf, wenn die CPU der SPS im Modus **KONSTANTE ZYKLUSDAUER** arbeitet und erkennt, dass die Zyklusdauer überschritten wurde. Die zusätzlichen Fehlerdaten enthalten die tatsächliche Zyklusdauer in den ersten beiden Byte und den Programmnamen in den nächsten acht Byte. Die Fehlerwirkung dieser Gruppe ist **Diagnose**.

Beseitigung:	(1) Wert für konstante Zyklusdauer erhöhen.
	(2) Anwenderprogramm verkürzen.

Anwendungsfehler

Die Fehlergruppe **Anwendungsfehler** tritt auf, wenn die CPU der SPS einen Fehler im Anwenderprogramm erkennt. Die Fehlerwirkung dieser Gruppe ist **Diagnose**, sofern es sich bei dem Fehler nicht um "Unterprogrammschachtelung überschritten" handelt. In diesem Fall ist die Fehlerwirkung **fatal**.

Fehlercode:	7
Name:	Unterprogrammschachtelung überschritten
Beschreibung:	Unterprogramme können bis zu einer Tiefe von 8 Ebenen geschachtelt werden. Ein Unterprogramm kann ein weiteres Unterprogramm aufrufen, das wieder ein Unterprogramm aufrufen kann, bis eine Schachtelungstiefe von 8 erreicht ist.
Beseitigung:	Programm so ändern, dass Schachtelungstiefe von 8 nicht überschritten wird.
Fehlercode:	1B
Name:	COMMREQ wurde wegen Speichereinschränkung bei der SPS nicht bearbeitet
Beschreibung:	Kommunikationsanforderungen ohne Wartezeit können in der Warteschlange schneller eingetragen werden, als sie verarbeitet werden können (z.B. eine pro Zyklus). Sammeln sich in einer solchen Situation so viele Kommunikationsanforderungen an, dass die SPS weniger als das Speicherminimum zur Verfügung hat, wird die Kommunikationsanforderung zum Fehler erklärt und nicht bearbeitet.
Beseitigung:	Weniger Kommunikationsanforderungen senden oder die im System ausgetauschte Mail auf andere Weise reduzieren.
Fehlercode:	5A
Name:	Anforderung anwenderseitige Abschaltung
Beschreibung:	Die SPS-Betriebssoftware (Funktionsblöcke) erzeugt diesen informatorischen Alarm, wenn im Anwenderprogramm die Dienstanforderung #13 (SPS abschalten) bearbeitet wird.
Beseitigung:	Keine. Informatorischer Alarm.

Anwenderprogramm fehlt

Die Fehlergruppe **Anwenderprogramm fehlt** tritt auf, wenn die CPU der SPS angewiesen wird, von **STOP** auf **RUN** zu gehen oder Daten in die SPS zu speichern, und in der SPS kein Anwenderprogramm vorhanden ist. Die CPU der SPS erkennt das Fehlen eines Anwenderprogramms beim Einschalten. Die Fehlerwirkung dieser Gruppe ist **informatorisch**.

Beseitigung:	Anwenderprogramm vor Übergang in **RUN-Modus laden.**

Verstümmeltes Anwenderprogramm beim Einschalten

Die Fehlergruppe **Verstümmeltes Anwenderprogramm beim Einschalten** tritt auf, wenn die CPU der SPS einen Fehler im Anwender-RAM erkennt. Die CPU der SPS bleibt solange in **STOP**-Modus, bis Anwenderprogramm und Konfigurationsdatei fehlerfrei geladen sind. Die Fehlerwirkung dieser Gruppe ist **fatal**.

Fehlercode:	1
Name:	Verstümmeltes Anwender-RAM beim Einschalten
Beschreibung:	Die SPS-Betriebssoftware erzeugt dieses Fehlermeldung, wenn sie beim Einschalten einen Defekt im Anwender-RAM erkennt.
Beseitigung:	(1) Konfigurationsdatei, Anwenderprogramm und Referenzen (falls vorhanden) neu laden. (2) Batterie der SPS CPU auswechseln. (3) Erweiterungsspeichermodul auf der SPS CPU auswechseln. (4) SPS CPU auswechseln.
Fehlercode:	2
Name:	Unzulässiger Boolescher Operationscode erkannt
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn sie im Anwenderprogramm eine fehlerhafte Anweisung erkennt.
Beseitigung:	(1) Anwenderprogramm und Referenzen (falls vorhanden) neu speichern. (2) Erweiterungsspeichermodul auf der SPS CPU auswechseln. (3) SPS CPU auswechseln.

Passwortfehler

Die Fehlergruppe **Passwortfehler** tritt auf, wenn die CPU der SPS eine Aufforderung zum Wechsel auf eine neue Privilegebene erhält und das mit der Anforderung eingegangene Passwort für diese Ebene nicht gilt. Die Fehlerwirkung dieser Gruppe ist **informatorisch**.

Beseitigung:	Anforderung mit richtigem Passwort wiederholen.

CPU-Systemsoftwarefehler in der SPS

Fehler in der Fehlergruppe **CPU-Systemsoftwarefehler** werden von der Betriebssoftware der CPU in der SPS Series 90-30, 90-20 oder Micro erzeugt. Sie können im Systembetrieb an unterschiedlichen Punkten auftreten. Tritt ein **fataler** Fehler auf, geht die CPU **sofort** in einen speziellen **FEHLER-ZYKLUS**-Modus. Es ist keinerlei Aktivität erlaubt, solange die SPS in diesem Modus ist. Sie können diesen Zustand nur aufheben, indem Sie die Betriebsspannung der SPS aus- und wieder einschalten. Die Fehlerwirkung dieser Gruppe ist **fatal**.

Fehlercode:	1 bis B
Name:	Anwenderspeicher konnte nicht belegt werden
Beschreibung:	Die SPS-Betriebssoftware (Speicherverwaltung) erzeugt diesen Fehler, wenn die Software die Speicherverwaltung auffordert, einen oder mehrere Blöcke aus dem Anwender-RAM zu belegen oder freizugeben, die unzulässig waren. Diese Fehler dürfen in einem Produktionssystem *nicht* vorkommen.
Beseitigung:	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
Fehlercode:	D
Name:	Kein Systemspeicher verfügbar
Beschreibung:	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn ihre Anforderung nach einen Systemspeicherblock von der Speicherverwaltung abgelehnt wurde, weil im Systemspeicher kein Platz frei ist. Tritt der Fehler während der Ausführung eines DOIO-Funktionsblocks auf, ist er *informatorisch*. Tritt er bei der Einschaltinitialisierung oder automatischen Konfiguration auf, ist er jedoch *fatal*.
Beseitigung:	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
Fehlercode:	E
Name:	Systemspeicher konnte nicht freigegeben werden
Beschreibung:	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn ihre Anforderung nach Freigabe eine Blocks im Systemspeicher nicht zu dem gewünschten Ergebnis geführt hat. Dieser Fehler kann nur während der Ausführung eines DOIO-Funktionsblocks auftreten.
Beseitigung:	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.
Fehlercode:	10
Name:	Unzulässige Abfrageanforderung von E/A-Zyklussteuerung
Beschreibung:	Die SPS-Betriebssoftware (E/A-Zyklussteuerung) erzeugt diesen Fehler, wenn weder Betriebssystem noch DOIO-Funktionsblockabfrage eine vollständige oder teilweise Abfrage der E/A fordern. Dies darf in einem Produktionssystem *nicht* vorkommen.
Beseitigung:	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
Fehlercode:	13
Name:	SPS-Betriebssoftwarefehler
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Dieser Fehler darf in einem Produktionssystem *nicht* vorkommen.
Beseitigung:	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.

Fehlercode:	14, 27
Name:	Verstümmelter SPS-Programmspeicher
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Diese Fehler dürfen in einem Produktionssystem *nicht* vorkommen.
Beseitigung:	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.
Fehlercode:	27 bis 4E
Name:	SPS-Betriebssoftwarefehler
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn bestimmte Probleme bei der SPS-Betriebssoftware auftreten. Diese Fehler dürfen in einem Produktionssystem *nicht* vorkommen.
Beseitigung:	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.
Fehlercode:	4F
Name:	Kommunikationsausfall
Beschreibung:	Die SPS-Betriebssoftware (Bedienanforderungsprozessor) erzeugt diesen Fehler, wenn sie versucht, auf eine Anforderung zu antworten, die Rückwandplatinen-Kommunikation erfordert, und dabei eine ablehnende Antwort erhält.
Beseitigung:	(1) Überprüfen Sie den Bus auf außergewöhnliche Aktivitäten. (2) Wechseln Sie das intelligente Zusatzmodul aus, an das die Anforderung gerichtet war.
Fehlercode:	50, 51, 53
Name:	Systemspeicherfehler
Beschreibung:	Die SPS-Betriebssoftware erzeugt diese Fehler, wenn ihre Anforderung nach einem Systemspeicherblock von der Speicherverwaltung abgelehnt wurde, da kein Speicher verfügbar ist oder der Speicher Fehler enthält.
Beseitigung:	(1) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten. (2) Führen Sie die Korrekturmaßnahmen für Speicherfehler durch.
Fehlercode:	52
Name:	Fehler bei Kommunikation über Rückwandplatine
Beschreibung:	Die SPS-Betriebssoftware (Bedienanforderungsprozessor) erzeugt diesen Fehler, wenn sie versucht, auf eine Anforderung zu antworten, die Rückwandplatinen-Kommunikation erfordert, und dabei eine ablehnende Antwort erhält.
Beseitigung:	(1) Überprüfen Sie den Bus auf außergewöhnliche Aktivitäten. (2) Wechseln Sie das intelligente Zusatzmodul aus, an das die Anforderung gerichtet war. (3) Überprüfen Sie, ob das Parallelkabel des Programmiergerätes richtig angeschlossen ist.
Fehlercode:	Alle übrigen.
Name:	Interner Systemfehler der SPS CPU
Beschreibung:	Ein interner Fehler ist aufgetreten, der in einem Produktionssystem **nicht** vorkommen darf.
Beseitigung:	Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.

Datenübertragungsfehler beim Speichern

Die Fehlergruppe **Datenübertragungsfehler beim Speichern** tritt während des Speicherns von Programmblöcken oder anderen Daten zur SPS auf. Der Strom der Befehle und Daten zum Speichern von Programmblöcken und Daten beginnt mit einem speziellen Sequenzanfangsbefehl und endet mit einem Sequenzendebefehl. Wird der Datenverkehr mit dem Programmiergerät, das den Speichervorgang durchführt, unterbrochen oder tritt ein anderer Fehler auf, der das Laden beendet, wird dieser Fehler eingetragen. Solange der Fehler im System ansteht, geht die Steuerung nicht in den **RUN**-Modus.

Dieser Fehler wird beim Einschalten *nicht* automatisch gelöscht. Der Anwender muss einen speziellen Befehl zum Löschen des Zustands eingeben. Die Fehlerwirkung dieser Gruppe ist **fatal**.

Beseitigung:	Fehler löschen und erneut versuchen, Programm oder Konfigurationsdatei zu laden.

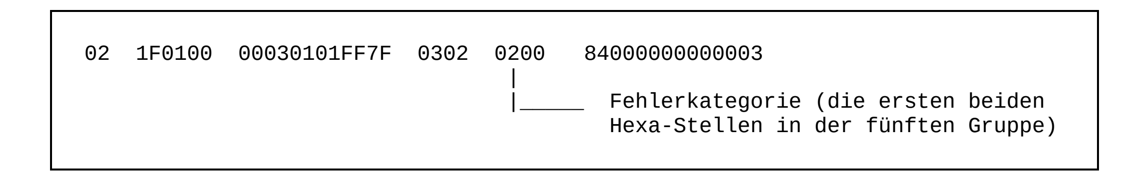
Abschnitt 3: E/A-Fehlertabelle - Erläuterungen

Die E/A-Fehlertabelle enthält Fehlerdaten in drei Klassifizierungen:

- Fehlerkategorie.
- Fehlertyp.
- Fehlerbeschreibung.

Die auf der nächsten Seite beschriebenen Fehler besitzen zwar eine Fehlerkategorie, haben aber weder Fehlertyp noch Fehlergruppe.

Jede Fehlererläuterung enthält eine Fehlerbeschreibung und Anweisungen, wie der Fehler behoben werden kann. Zahlreiche Fehlerbeschreibungen haben mehrere Ursachen. In diesen Fällen wird der Fehlercode, der bei den über CTRL-F zugänglichen zusätzlichen Fehlerinformationen angezeigt wird, zur Unterscheidung unterschiedlicher Fehlerzustände mit gleicher Fehlerbeschreibung benutzt. (Weitere Angaben zur Verwendung von CRTL-F finden Sie in Anhang B, Fehlertabellen interpretieren, in diesem Handbuch.) Die Fehlerkategorie besteht aus den ersten beiden Hexadezimalstellen der fünften Zahlengruppe (siehe nachstehendes Beispiel).



```
02  1F0100  00030101FF7F  0302  0200   84000000000003
                                 |
                                 |_____  Fehlerkategorie (die ersten beiden
                                         Hexa-Stellen in der fünften Gruppe)
```

Die nachstehende Tabelle soll Ihnen helfen, schnell eine bestimmte E/A-Fehlerbeschreibung in diesem Abschnitt zu finden. Die einzelnen Einträge sind in der Reihenfolge aufgeführt, wie sie auf dem Programmiergeräte-Bildschirm erscheinen.

Verlorenes E/A-Modul

Die Fehlerkategorie **Verlorenes E/A-Modul** bezieht sich auf diskrete und analoge Modell 30 E/A-Module. Mit dieser Kategorie sind weder Fehlertypen noch Fehlerbeschreibungen verknüpft. Die Fehlerwirkung ist **Diagnose**.

Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler wenn sie merkt, dass ein Modell 30 E/A-Modul nicht mehr auf Befehle von der SPS-CPU reagiert oder wenn ein leerer Steckplatz laut Konfigurationsdatei ein Modul enthalten müsste.
Beseitigung:	(1) Wechseln Sie das Modul aus. (2) Korrigieren Sie die Konfigurationsdatei. (3) Rufen Sie die SPS-Fehlertabelle auf dem Bildschirm des Programmiergerätes auf. Nehmen Sie dann Kontakt mit der Serviceabteilung von GE Fanuc auf und nennen Sie alle im Fehlereintrag enthaltenen Daten.

Hinzugefügtes E/A-Modul

Die Fehlerkategorie **Hinzugefügtes E/A-Modul** bezieht sich auf diskrete und analoge Modell 30 E/A-Module. Mit dieser Kategorie sind weder Fehlertypen noch Fehlerbeschreibungen verknüpft. Die Fehlerwirkung ist **Diagnose**.

Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn ein zuvor fehlerhaftes E/A-Modul wieder zum normalen Betrieb zurückkehrt.
Beseitigung:	(1) Keine Maßnahmen erforderlich, wenn das Modul entfernt oder ausgewechselt wurde oder die Versorgungsspannung zum dezentralen Chassis aus- und wieder eingeschaltet wurde. (2) Aktualisieren Sie die Konfiguration oder entfernen Sie das Modul.
Beschreibung:	Die SPS-Betriebssoftware erzeugt diesen Fehler, wenn sie ein Modell 30 E/A-Modul in einem Steckplatz erkennt, der laut Konfigurationsdatei leer sein sollte.
Beseitigung:	(1) Entfernen Sie das Modul (es kann z.B. im falschen Steckplatz stecken). (2) Aktualisieren und speichern Sie die Konfigurationsdatei so, dass sie das überzählige Modul enthält.

Kapitel 4 Relaisfunktionen

In diesem Kapitel wird die Verwendung von Kontakten, Spulen und Verbindungen in den Strompfaden eines Kontaktplanprogramms erläutert.

Kontakte

Mit einem Kontakt wird der Zustand einer Referenz überwacht. Der Stromfluss über einen Kontakt hängt vom Zustand der überwachten Referenz und vom Kontakttyp ab. Eine Referenz ist EIN, wenn ihr Zustand 1 ist und AUS, wenn ihr Zustand 0 ist.

Tabelle 4-1. Kontakttypen

Kontakttyp	Symbol	Kontakt schaltet Stromfluss nach rechts durch,
Schließerkontakt	—| |—	wenn die Referenz EIN ist.
Öffnerkontakt	—|/|—	wenn die Referenz AUS ist.
Fortsetzkontakt	<+>———	wenn die vorhergehende Fortsetzspule EIN ist.

Spulen

Über Spulen werden diskrete Referenzen gesteuert. Der Stromfluss zu einer Spule muss über eine Bedingungslogik gesteuert werden. Spulen verursachen direkt eine Aktion, sie geben keinen Stromfluss nach rechts weiter. Soll als Ergebnis eines Kontaktzustandes ein weiterer Programmteil durchlaufen werden, dann muss für diese Spule eine interne Referenz oder eine Kombination Fortsetzmerker/-kontakt verwendet werden.

Spulen stehen immer ganz rechts in einer Programmzeile. In einem Strompfad sind bis zu acht parallele Spulen möglich.

Die verwendeten Spulentypen hängen davon ab, welche Aktion das Programm durchführen soll. Die Zustände remanenter Spulen bleiben bei einem Ausfall der Versorgungsspannung oder beim Umschalten von **STOP**- in **RUN**-Modus erhalten. Die Zustände nicht remanenter Spulen gehen bei der Spannungsrückkehr nach einem Ausfall der Versorgungsspannung oder beim Umschalten von **STOP**- in **RUN**-Modus auf Null.

Tabelle 4-2. Spulentypen

Spulentyp	Anzeige	Strom zur Spule	Ergebnis
Schließer	—()—	EIN AUS	Schaltet Referenz EIN. Schaltet Referenz AUS.
Negiert	—(/)—	EIN AUS	Schaltet Referenz AUS. Schaltet Referenz EIN.
Remanent	—(M)—	EIN AUS	Schaltet Referenz EIN, remanent. Schaltet Referenz AUS, remanent.
Negiert Remanent	—(/M)—	EIN AUS	Schaltet Referenz AUS, remanent. Schaltet Referenz EIN, remanent.
Positiver Übergang	—(↑)—	AUS→EIN	Ist die Referenz AUS, geht sie für einen Zyklus auf EIN.
Negativer Übergang	—(↓)—	EIN←AUS	Ist die Referenz AUS, geht sie für einen Zyklus auf EIN.
SET	—(S)—	EIN AUS	Schaltet Referenz EIN, bis sie von —(R)— auf AUS gesetzt wird. Spulenzustand unverändert.
RESET	—(R)—	EIN AUS	Schaltet Referenz AUS, bis sie von —(S)— auf EIN gesetzt wird. Spulenzustand unverändert.
SET remanent	—(SM)—	EIN AUS	Schaltet Referenz EIN, bis sie von —(RM)— auf AUS gesetzt wird; remanent. Spulenzustand unverändert.
Remanent RESET	—(RM)—	EIN AUS	Schaltet Referenz AUS, bis sie von —(SM)— auf EIN gesetzt wird; remanent. Spulenzustand unverändert.
Fortsetzspule	——<+>	EIN AUS	Schaltet nächsten Fortsetzkontakt EIN. Schaltet nächsten Fortsetzkontakt AUS.

Schließerkontakt —| |—

Ein Schließerkontakt arbeitet als Schalter, der den Stromfluss weiterleitet, wenn die zugehörige Referenz EIN (1) ist.

Öffnerkontakt —|/|—

Ein Öffnerkontakt arbeitet als Schalter, der den Stromfluss weiterleitet, wenn die zugehörige Referenz AUS (0) ist.

Beispiel

Das folgende Beispiel zeigt einen Strompfad mit 10 Elementen, denen die symbolischen Adressen E1 bis E10 zugeordnet sind. Die Spule E10 ist dann durchgeschaltet, wenn die Referenzen E1, E2, E5, E6 und E9 EIN und die Referenzen E3, E4, E7 und E8 AUS sind.

```
|
|  E1       E2       E3       E4       E5       E6       E7       E8       E9       E10
|——| |—————| |—————|/|—————|/|—————| |—————| |—————|/|—————|/|—————| |—————( )—
|
```

Spule —( )—

Empfängt eine Spule Stromfluss, schaltet sie die zugehörige diskrete Referenz auf EIN. Eine Spule ist nicht remanent und kann daher nicht zusammen mit Systemzustandsreferenzen (%SA, %SB, %SC oder %G) verwendet werden.

Beispiel

Im folgenden Beispiel wird die Spule E3 durchgeschaltet, wenn Referenz E1 EIN und Referenz E2 AUS ist.

```
|
|  E1       E2                                                                     E3
|——| |—————|/|—————————————————————————————————————————————————————————————————————( )—
|
```

Negierte Spule —(/)—

Eine negierte Spule schaltet eine Referenz auf EIN, wenn sie keinen Stromfluss empfängt. Sie ist nicht remanent und kann daher nicht zusammen mit Systemzustandsreferenzen (%SA, %SB, %SC oder %G) verwendet werden.

Beispiel

Im folgenden Beispiel wird die Spule E3 durchgeschaltet, wenn die Referenz E1 AUS ist.

```
|
|  E1                                                                  E2
|--| |-----------------------------------------------------------------(/)-
|
|  E2                                                                  E3
|--| |-----------------------------------------------------------------( )-
|
```

Remanente Spule —(M)—

Wie bei einer Spule wird bei der remanenten Spule eine diskrete Referenz eingeschaltet, wenn die Spule Stromfluss empfängt. Der Zustand der remanenten Spule wird über einen Stromausfall hinweg erhalten. Sie kann daher nicht mit Referenzen aus streng nicht-remanenten Speicherbereichen (%T) verwendet werden.

Negierte remanente Spule —(/M)—

Die negierte remanente Spule schaltet eine diskrete Referenz EIN, wenn sie keinen Stromfluss empfängt. Der Zustand der negierten remanenten Spule wird über einen Stromausfall hinweg erhalten. Sie kann daher nicht mit Referenzen aus streng nicht-remanenten Speicherbereichen (%T) verwendet werden.

Spule für positive Flanken —(↑)—

Ist die mit einer Spule für positive Flanken verbundene Referenz AUS, dann wird sie durchgeschaltet, wenn die Spule Stromfluss empfängt. Alle mit dieser Spule verknüpften Kontakte wechseln einen SPS-Zyklus lang ihren Zustand. (Wird der Strompfad, der die Spule enthält, bei den nachfolgenden Zyklen übersprungen, dann bleibt die Spule EIN.) Diese Spule kann als Wischrelais verwendet werden.

Um die Wischrelaisfunktion der Spule zu erhalten darf jede Referenz im Anwenderprogramm nur einmal als Übergangsspule verwendet werden.

Übergangsspulen können mit Referenzen aus remanenten und nicht-remanenten Speicherbereichen (%Q, %M, %T, %G, %SA, %SB oder %SC) verwendet werden.

Spulen für negative Flanken —(↓)—

Ist die mit einer Spule für negative Flanken verbundene Referenz AUS, wenn der Stromfluss zur Spule ***unterbrochen*** wird, wird sie durchgeschaltet, bis die Spule erneut bearbeitet wird. Alle mit dieser Spule verbundenen Kontakte wechseln einen Zyklus lang ihren Zustand.

Um die Wischrelaisfunktion der Spule zu erhalten darf jede Referenz im Anwenderprogramm nur einmal als Übergangsspule verwendet werden.

Übergangsspulen können mit Referenzen aus remanenten und nicht-remanenten Speicherbereichen (%Q, %M, %T, %G, %SA, %SB oder %SC) verwendet werden.

Beispiel

Schaltet bei dem folgenden Beispiel die Referenz E1 von AUS nach EIN, dann fließt Strom zu den Spulen E2 und E3 und E2 wird für die Dauer eines logischen Zyklus durchgeschaltet. Schaltet E1 von EIN nach AUS, dann wird der Stromfluss zu E2 und E3 unterbrochen, und die Spule E3 wird für die Dauer eines Zyklus durchgeschaltet.

```
|
|    E1                                                     E2
|——| |—————————————————————————————————————————————————————(↑)—
|
|    E1                                                     E3
|——| |—————————————————————————————————————————————————————(↓)—
|
```

SET-Spule —(S) —

SET und RESET sind nicht-remanente Spulen, die den Zustand einer Referenz (z.B. E1) auf EIN oder AUS halten ("verriegeln") können. Wird der Stromfluss zu einer SET-Spule geschaltet, bleibt ihre Referenz solange EIN (unabhängig davon, ob die Spule selbst Stromfluss empfängt), bis diese Referenz durch eine andere Spule rückgesetzt wird.

SET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbit für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2, "Systembetrieb”).

RESET-Spule —(R)—

Empfängt die RESET-Spule Stromfluss, wird eine diskrete Referenz auf AUS geschaltet. Diese Referenz bleibt solange AUS, bis sie durch eine andere Spule wieder eingeschaltet wird. Die im Programm zuletzt bearbeitete Spule einer SET-/RESET-Kombination hat jeweils Vorrang.

RESET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbit für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2, "Systembetrieb”).

Beispiel

Beim folgenden Beispiel wird die Spule E1 durchgeschaltet, wenn Referenz E2 oder E6 EIN ist. Die Spule E1 wird abgeschaltet, wenn Referenz E5 oder E3 EIN ist.



Hinweis

Bei Spulenüberprüfung EINFACH können Sie eine bestimmte %M- oder %Q-Referenz zwar mit nur einer Spule benutzen, aber Sie können sie gleichzeitig mit einer SET- und einer RESET-Spule einsetzen. Bei Spulenüberprüfung WARNEN MEHRFACH oder MEHRFACH können Sie die einzelnen Referenzen mit mehreren Spulen, SET-Spulen und RESET-Spulen verwenden. Bei Mehrfachbenutzung könnte eine Referenz entweder durch eine SET-Spule oder eine normale Spule EIN geschaltet werden und durch eine RESET-Spule oder eine normale Spule AUS geschaltet werden.

Remanente SET-Spule —(SM)—

Die Funktion remanenter SET- und RESET-Spulen ist ähnlich wie die der normalen SET- und RESET-Spulen, ihre Zustände bleiben jedoch bei einem Ausfall der Versorgungsspannung oder beim Umschalten der SPS von **STOP**- in **RUN**-Modus erhalten. Eine remanente SET-Spule schaltet eine diskrete Referenz durch, wenn die Spule Stromfluss empfängt. Die Referenz bleibt durchgeschaltet, bis sie durch eine remanente RESET-Spule wieder rückgesetzt wird.

Remanente SET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbit für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2, "Systembetrieb”).

Remanente RESET-Spule —(RM)—

Empfängt die remanente RESET-Spule Stromfluss, dann wird eine diskrete Referenz auf AUS geschaltet. Diese Referenz bleibt solange AUS, bis sie durch eine remanente SET-Spule wieder eingeschaltet wird. Der Zustand dieser Spule bleibt bei einem Ausfall der Versorgungsspannung oder beim Umschalten von **STOP**- in **RUN**-Modus erhalten.

Remanente RESET-Spulen schreiben ein undefiniertes Ergebnis in das Übergangsbit für die angegebene Referenz (siehe "Transitionen und Overrides" in Kapitel 2, "Systembetrieb”).

Verbindungen

Über Horizontal- und Vertikalverbindungen werden in einer Kontaktplanzeile Funktionen miteinander verbunden. Sie vervollständigen in einer Programmzeile den Weg für den Stromfluss von links nach rechts.

Hinweis

Sie können eine Horizontalverbindung nicht dazu benutzen, eine Funktion oder eine Spule an die linke Stromschiene anzuschließen. Um eine Funktion in jedem Zyklus aufzurufen, können Sie jedoch %S7, das Systembit ALW_ON (immer EIN), mit einem an der Stromschiene angeschlossenen Schließerkontakt verwenden.

Beispiel

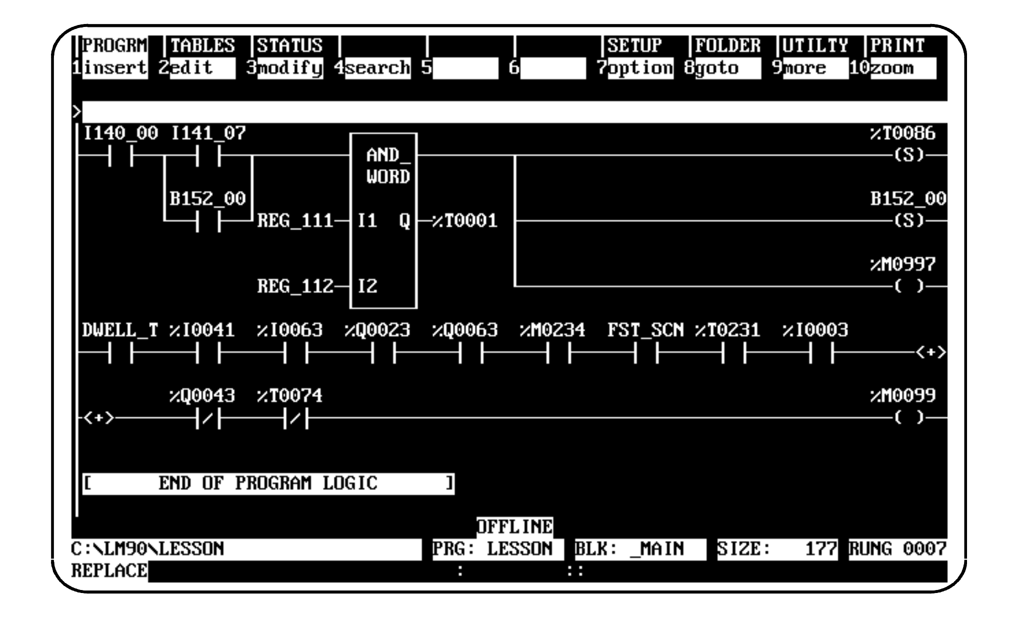
Im folgenden Beispiel werden zwei Horizontalverbindungen dazu verwendet, die Kontakte E2 und E5 miteinander zu verbinden. Die Verknüpfung der Kontakte E3, E6, E7, E8 und E9 mit E2 wird über eine Vertikalverbindung hergestellt.

```
|
|
|  E2                       E5                                                E1
|——| |——+———————+———————+——| |——————————————————————————————————————————————(/)—
|                       |
|                       |
|  E3      E6      E7   |
|——| |——+——|/|——+——| |——+
|        |       |      |
|        |       |      |
|        |  E8   |  E9  |
|        +——| |——+——| |——+
|
```

Fortsetzspulen (———<+>) und -kontakte (<+>———)

Mit Fortsetzspulen (———<+>) und Fortsetzkontakten (<+>———) wird die Kontaktplanlogik über die Grenze von zehn Spalten hinweg fortgeführt. Der Zustand der zuletzt ausgeführten Fortsetzspule wird als Stromflusszustand für den als nächstes ausgeführten Fortsetzkontakt verwendet. Wird im Stromfluss keine Fortsetzspule vor einem Fortsetzkontakt ausgeführt, dann ist der Zustand dieses Kontakts stromlos. Der Zustand des Fortsetzkontaktes wird auf Null gesetzt, wenn die SPS von **Stop** auf **Run** geht und es gibt nur dann Stromfluss, wenn die Übergangsspule nach dem Übergang in **Run**-Modus gesetzt wird.

Pro Strompfad sind nur eine Fortsetzspule und ein Fortsetzkontakt zulässig. Der Fortsetzkontakt muss in Spalte 1 stehen, die Fortsetzspule in Spalte 10. Die nachstehende Abbildung zeigt ein Beispiel für eine Fortsetzspule und einen Fortsetzkontakt:



Kapitel 5 Zeitgeber und Zähler

In diesem Kapitel wird die Verwendung von Zeitgliedern zur Einschaltverzögerung und Zeiterfassung sowie Auf- und Abwärtszählern beschrieben. Die mit diesen Funktionen verknüpften Daten sind nullspannungssicher.

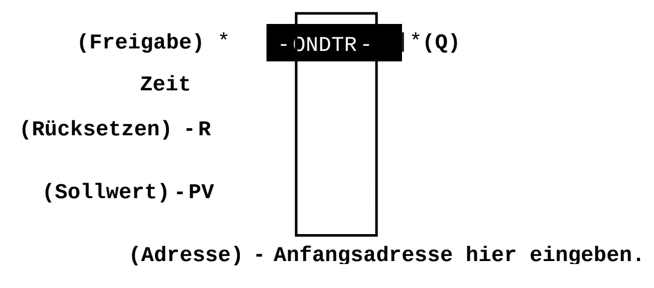
Abkürzung	Funktion	Seite
ONDTR	Remanenter Einschaltverzögerungs-Zeitgeber	5-3
TMR	Einfacher Einschaltverzögerungs-Zeitgeber	5-6
OFDT	Ausschaltverzögerungs-Zeitgeber	5-9
UPCTR	Aufwärtszähler	5-12
DNCTR	Abwärtszähler	5-13

Für Zeitgeber und Zähler benötigte Funktionsblockdaten

Jeder Zeitgeber und jeder Zähler belegt im %R-Speicher drei Worte (Register), um die folgenden Daten abzuspeichern:

Istwert (CV)	Wort 1
Sollwert (PV)	Wort 2
Steuerwort	Wort 3

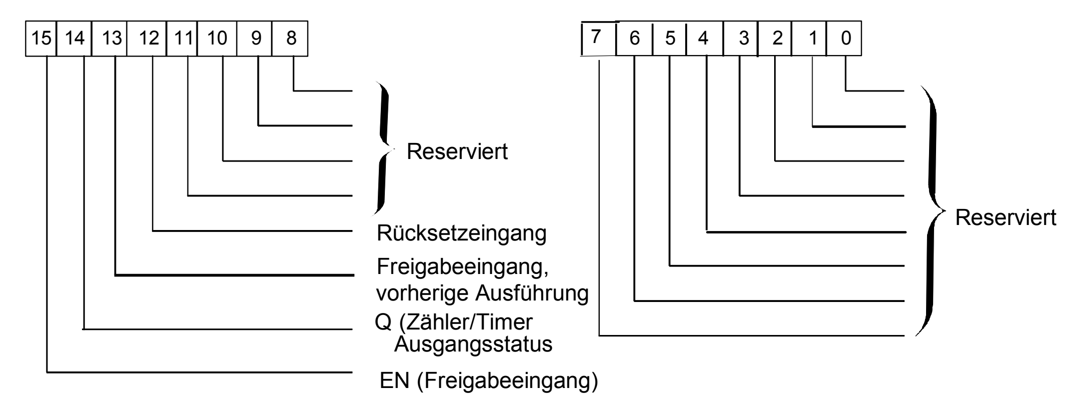
Bei der Eingabe eines Zeitgebers oder Zählers müssen Sie immer eine Anfangsadresse für diese drei Worte (Register) unmittelbar unterhalb der Funktionsgraphik eingeben. Zum Beispiel:



Hinweis

Verwenden Sie für die 3-Wort-Zeitgeber/Zählerblöcke keine aufeinanderfolgenden Register. Logicmaster prüft ***nicht***, ob sich Registerblöcke überschneiden. Zeitgeber und Zähler funktionieren nicht, wenn Sie mit dem Istwert eines Blocks den Sollwert des vorherigen Blocks überschreiben.

Im Steuerwort werden die Zustände der Booleschen Ein- und Ausgänge des zugehörigen Funktionsblocks in folgendem Format abgelegt:



Bits 0 bis 11 werden für Zeitgebergenauigkeit eingesetzt. Bits 0 bis 11 werden für Zähler nicht benutzt.

Hinweis

Lassen Sie Vorsicht walten, wenn Sie für PV (Sollwert) und das zweite Wort in dem Block von drei Worten die gleiche Adresse verwenden. Ist PV keine Konstante, werden PV und das zweite Wort normalerweise auf unterschiedliche Adressen gesetzt. Bei einigen Anwendungen wird die zweite Wortadresse für PV benutzt (z.B. %R0102, wenn der untere Datenblock bei %R0101 beginnt). Hierdurch kann das Anwenderprogramm den Wert von PV bei laufendem Zeitgeber oder Zähler ändern. Das erste Istwert- oder das dritte Steuerwort können zwar von Anwenderprogrammen gelesen werden, das Anwenderprogramm kann aber diese Werte nicht überschreiben, da sonst die Funktion nicht funktioniert.

Spezieller Hinweis für bestimmte Bitoperationen

Beim Einsatz der Funktionen Bit testen, Bit auf 1 setzen, Bit auf 0 setzen oder Bitposition, werden die Bits von 1 bis 16 nummeriert, ***NICHT*** von 0 bis 15, wie dies das vorstehende Diagramm zeigt.

ONDTR

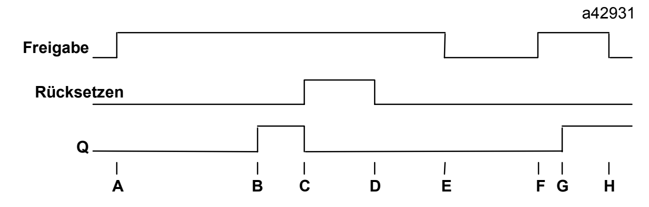
Ein remanenter Zeitgeber zur Einschaltverzögerung (ONDTR) erhöht seinen Wert, solange er Stromfluss empfängt, und hält den Wert, wenn der Stromfluss stoppt. Die Zeitzählung erfolgt dabei in Schritten von 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunden in einem Bereich zwischen 0 und 32.767 Zeiteinheiten. Dieser Zeitgeber ist nullspannungssicher, beim Einschalten wird er nicht automatisch initialisiert.

Die ONDTR-Funktion beginnt mit der Zeitzählung (Istwert), wenn sie erstmals Stromfluss empfängt. Der Istwert wird aktualisiert, wenn der Zeitgeber im Kontaktplanprogramm angetroffen wird.

Hinweis

Empfängt der gleiche Zeitgeber an mehreren Stellen (mit der gleichen Referenzadresse) während eines CPU-Zyklus Stromfluss, dann sind alle Zeitwerte gleich.

Der Ausgang Q wird durchgeschaltet, wenn der Istwert gleich dem Sollwert PV ist oder diesen übersteigt. Der Istwert im Zeitgeber wird bis zum Maximalwert erhöht, solange der Stromfluss zu dem Zeitglied besteht. Ist der Maximalwert erreicht, wird er gehalten und Ausgang Q bleibt unabhängig vom Zustand des Freigabeeingangs durchgeschaltet.



A = Das Freigabesignal (ENABLE) wird aktiv, der Zeitgeber beginnt mit der Zeitzählung.

B = Der Istwert erreicht den Sollwert (PV); Ausgang Q wird durchgeschaltet.

C = Das Rücksetzsignal (RESET) wird aktiv; Q wird abgeschaltet, der Istwert rückgesetzt.

D = Das Rücksetzsignal (RESET) wird zurückgenommen, der Zeitgeber beginnt wieder mit der Zählung.

E = Das Freigabesignal (ENABLE) wird zurückgenommen, die Zeitzählung gestoppt. Der Istwert bleibt erhalten.

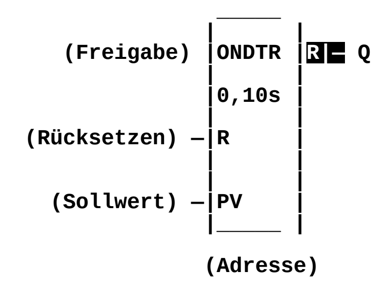
F = Das Freigabesignal (ENABLE) wird wieder aktiv, der Zeitgeber fährt mit der Zeitzählung fort.

G = Der Istwert erreicht den Sollwert (PV), der Ausgang Q wird durchgeschaltet. Der Zeitgeber fährt mit der Zeitzählung fort, bis entweder das Freigabesignal zurückgenommen wird, RESET aktiv wird oder der Istwert den Maximalwert erreicht.

H = Das Freigabesignal (ENABLE) wird zurückgenommen, die Zeitzählung wird angehalten.

Wird der Stromfluss zum Zeitgeber unterbrochen, dann wird die Erhöhung des Istwerts angehalten und der Istwert bleibt erhalten. War Ausgang Q stromführend, bleibt er durchgeschaltet. Die Erhöhung des Istwerts wird bei dem erhaltenen Wert fortgesetzt, wenn der Stromfluss zum Zeitgeber wieder einsetzt. Bei Stromfluss zum Rücksetzeingang (R) wird der Istwert auf Null gesetzt und der Ausgang Q wird abgeschaltet. Bei SPS-Modellen 35x und 36x wird das Ausgangssignal zurückgenommen, wenn die Freigabe zum ONDTR auf L-Pegel ist, PV=0 und der Rücksetzeingang R

Stromfluss empfängt. Bei den SPS-Modellen 311 bis 341 ist jedoch unter den gleichen Voraussetzungen der Ausgang aktiv.



Parameter

Parameter	Beschreibung
Adresse	Die ONDTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: • Istwert (CV) = Wort 1. • Sollwert (PV) = Wort 2. • Steuerwort = Wort 3. Bei der Eingabe der ONDTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. **Hinweis:** Verwenden Sie diese Adresse nicht bei anderen Anweisungen. **Vorsicht:** Adressüberschneidungen führen zu Störungen im Zeitgeberbetrieb.
Freigabe	Der aktuelle Zeitgeberwert wird erhöht, wenn der Freigabeeingang Stromfluss erhält.
R	Der Istwert wird auf Null gesetzt, wenn Eingang R Stromfluss erhält.
PV	Der Wert von PV wird in den Sollwert des Zeitgebers kopiert, wenn der Zeitgeber freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert größer als oder gleich dem Sollwert ist.
Zeit	Die Zeitschritte für das untere Bit des Sollwerts PV sind Zehntel- (0,1), Hundertstel- (0,01) oder Tausendstelsekunden (0,001).

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

• Zulässiger Datentyp oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

In dem folgenden Beispiel wird ein remanenter Einschaltverzögerungs-Zeitgeber dazu verwendet, ein Signal (%Q0011) zu erzeugen, das 8,0 Sekunden nach Einschalten von %Q0010 aktiv wird und das abgeschaltet wird, wenn %Q0010 abschaltet.

```
|          _____
| %Q0010  |     |                                                       %Q0011
|--| |----|ONDTR|-------------------------------------------------------( )-
|         | 0.1s|
| %Q0010  |     |
|--|/|----|R    |
|         |     |
|         |     |
| CONST --|PV   |
| +00080  |_____|
|
|        %R0004
```

TMR

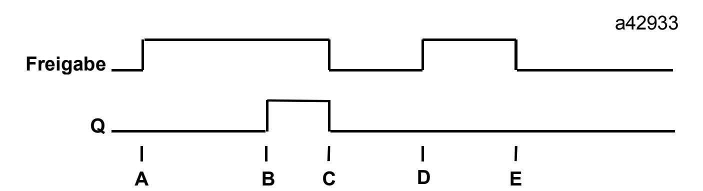
Der einfache Zeitgeber zur Einschaltverzögerung (TMR) erhöht seinen Wert, solange er Stromfluss empfängt, und setzt den Wert auf Null, wenn der Stromfluss stoppt. Die Zeitzählung erfolgt dabei in Schritten von 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunden Der Bereich liegt zwischen 0 und +32.767 Zeiteinheiten (d.h. 0,001 bis 3.276,7 Sekunden). Dieser Zeitgeber ist nullspannungssicher, beim Einschalten wird er nicht automatisch initialisiert.

Die TMR-Funktion beginnt mit der Zeitzählung (Istwert), wenn sie Stromfluss empfängt. Der Istwert wird aktualisiert, wenn der Zeitgeber im Kontaktplanprogramm angetroffen wird, und gibt so die gesamte Zeitdauer an, die der Zeitgeber seit dem letzten Rücksetzen freigegeben war.

Hinweis

Empfängt der gleiche Zeitgeber an mehreren Stellen (mit der gleichen Referenzadresse) während eines CPU-Zyklus Stromfluss, dann sind alle Zeitwerte gleich.

Diese Aktualisierung dauert solange, wie die Freigabelogik EIN bleibt. Die Funktion schaltet den Stromfluss nach rechts durch, wenn der Istwert größer/gleich dem Sollwert (PV) ist. Der Zeitgeber erhöht den Istwert solange, bis der Maximalwert erreicht ist. Wechselt der Freigabeparameter von EIN nach AUS, stoppt der Zeitgeber und der Istwert wird auf Null rückgesetzt.



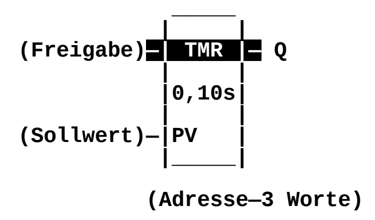
A = Das Freigabesignal (ENABLE) wird aktiv, der Zeitgeber beginnt mit der Zeitzählung.

B = Der Istwert erreicht den Sollwert (PV). Der Ausgang Q wird durchgeschaltet und der Zeitgeber fährt mit der Zeitzählung fort.

C = Das Freigabesignal (ENABLE) wird zurückgenommen und Ausgang Q abgeschaltet. Die Zeitzählung wird angehalten und der Istwert wird gelöscht.

D = Das Freigabesignal (ENABLE) wird aktiv, der Zeitgeber beginnt mit der Zeitzählung.

E = Das Freigabesignal (ENABLE) wird zurückgenommen ehe der Istwert den Sollwert (PV) erreicht. Q bleibt abgeschaltet, die Zeitzählung wird angehalten und auf Null gesetzt.



Parameter

Parameter	Beschreibung
Adresse	Die TMR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: • Istwert (CV) = = Wort 1. • Sollwert (PV) = Wort 2. • Steuerwort = Wort 3. Bei der Eingabe der TMR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. **Hinweis:** Verwenden Sie diese Adresse nicht bei anderen Anweisungen. **Vorsicht:** Adressüberschneidungen führen zu Störungen im Zeitgeberbetrieb.
Freigabe	Der aktuelle Zeitgeberwert wird erhöht, wenn der Freigabeeingang Stromfluss erhält. Der Istwert wird auf Null gesetzt und Q abgeschaltet, wenn die TMR-Funktion nicht freigegeben ist.
PV	Der Wert von PV wird in den Sollwert des Zeitgebers kopiert, wenn der Zeitgeber freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn TMR freigegeben und der Istwert größer als oder gleich dem Sollwert ist.

Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

• Zulässiger Datentyp oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Im folgenden Beispiel wird ein Zeitgeber zur Einschaltverzögerung (mit Adresse) TMRID dazu verwendet, die Einschaltdauer einer Spule DWELL zu steuern. Wird der Schließerkontakt (Taster) DO_DWL geschlossen, wird die Spule DWELL durchgeschaltet. Der Haltekontakt der Spule DWELL hält die Spule DWELL durchgeschaltet (nachdem der Kontakt DO_DWL losgelassen wurde) und startet auch den Zeitgeber TMRID. Wenn TMRID seinen Sollwert von einer halben Sekunde erreicht hat, schaltet die Spule REL durch und unterbricht dadurch den Haltezustand der Spule DWELL. Der Kontakt DWELL unterbricht den Stromfluss zu TMRID, setzt dessen Istwert zurück und schaltet die Spule REL ab. Der Kreis ist dann wieder für die nächste Betätigung des Kontaktes DO_DWL bereit.

```
|
| DO_DWL    REL                                                        DWELL
|--| |--+--|/|--------------------------------------------------------( )-
|       |
| DWELL |
|--| |--+
|          _____
| DWELL   |     |                                                       REL
|--| |----| TMR |------------------------------------------------------( )-
|         | 0.1s|
|         |     |
| CONST --|PV   |
| +00005  |_____|
|
|          TMRID
```

OFDT

Der Abschaltverzögerungs-Zeitgeber (OFDT) erhöht seinen Wert, wenn der Stromfluss abgeschaltet ist, und wird durch Stromfluss auf Null gesetzt. Die Zeitzählung erfolgt dabei in Schritten von 1/10 (Standardeinstellung), 1/100 oder 1/1000 Sekunden in einem Bereich zwischen 0 und 32.767 Zeiteinheiten. Dieser Zeitgeber ist nullspannungssicher, beim Einschalten wird er nicht automatisch initialisiert.

Empfängt der OFDT erstmalig Stromfluss, wird der Stromfluss nach rechts durchgeschaltet und der Istwert (PV) auf Null gesetzt. (Der OFDT benutzt Wort 1 (Register) als Speicherplatz für CV - siehe Abschnitt "Parameter:" auf der nächsten Seite.) Der Ausgang bleibt durchgeschaltet, solange die Funktion Stromfluss erhält. Wird der Stromfluss von links unterbrochen, gibt sie weiterhin Stromfluss nach rechts ab und der Zeitgeber beginnt mit der Zeitzählung im Istwert.

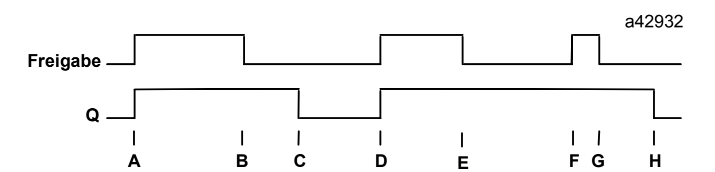
Hinweis

Empfängt der gleiche Zeitgeber an mehreren Stellen (mit der gleichen Referenzadresse) während eines CPU-Zyklus Stromfluss, dann sind alle Zeitwerte gleich.

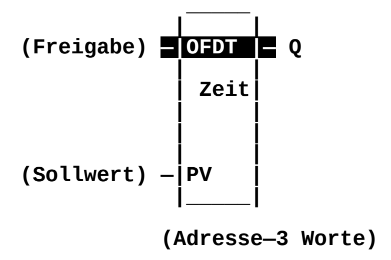
Der OFDT schaltet den Stromfluss nicht durch, wenn der Sollwert Null oder negativ ist.

Jedesmal, wenn die Funktion mit abgeschalteter Freigabelogik aufgerufen wird, wird der Istwert aktualisiert und gibt so die Zeit an, die seit dem Abschalten des Zeitgebers verstrichen ist. Erreicht der Istwert (CV) den Sollwert (PV), unterbricht die Funktion den Stromfluss nach rechts. Der Zeitgeber hält dann mit der Zeitzählung an — siehe Teil C im nachstehenden Ablaufdiagramm.

Der Sollwert wird auf Null rückgesetzt, wenn die Funktion wieder Stromfluss erhält.



A = Freigabesignal (ENABLE) und Q werden beide aktiv; der Zeitgeber wird rückgesetzt (CV = 0).
B = Das Freigabesignal (ENABLE) wird zurückgenommen, der Zeitgeber beginnt mit der Zählung.
C = CV erreicht PV; Q wird zurückgenommen; der Zeitgeber stoppt die Zeitzählung.
D = Das Freigabesignal (ENABLE) wird aktiv, der Zeitgeber wird rückgesetzt (CV = 0).
E = Das Freigabesignal (ENABLE) wird zurückgenommen, der Zeitgeber beginnt mit der Zählung.
F = Das Freigabesignal (ENABLE) wird aktiv, der Zeitgeber wird rückgesetzt (CV = 0).
G = Das Freigabesignal (ENABLE) wird zurückgenommen, der Zeitgeber beginnt mit der Zählung.
H = CV erreicht PV; Q wird zurückgenommen; der Zeitgeber stoppt die Zeitzählung.



Wird der OFDT in einem Programmblock verwendet, der ***nicht*** in jedem Zyklus aufgerufen wird, zählt der Zeitgeber die Zeit zwischen den Aufrufen des Programmblocks (sofern er nicht rückgesetzt wird). Das heißt, dass er funktioniert wie ein Zeitgeber in einem Programm mit einem viel langsameren Zyklus als der Zeitgeber im Hauptprogrammblock. Bei Programmblöcken, die für längere Zeit inaktiv sind, muss der Zeitgeber so programmiert werden, dass diese Aufholfunktion berücksichtigt wird. Wird zum Beispiel ein Zeitgeber in einem Programmblock rückgesetzt und wird der Programmblock vier Minuten lang nicht aufgerufen (ist inaktiv), sind bereits vier Minuten aufgelaufen, wenn der Programmblock aufgerufen wird. Wird der Zeitgeber nicht zuerst rückgesetzt, dann wird er bei der Freigabe mit diesen vier Minuten beaufschlagt.

Parameter

Parameter	Beschreibung
Adresse	Die OFDT-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: • Istwert (CV) = Wort 1. • Sollwert (PV) = Wort 2. • Steuerwort = Wort 3. Bei der Eingabe der OFDT-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. **Hinweis:** Verwenden Sie diese Adresse nicht bei anderen Anweisungen. **Vorsicht:** Adressüberschneidungen führen zu Störungen im Zeitgeberbetrieb.
Freigabe	Der aktuelle Zeitgeberwert wird erhöht, wenn der Freigabeeingang Stromfluss erhält.
Zeit	Die Zeitschritte für das untere Bit des Sollwerts PV sind Zehntel- (0,1), Hundertstel- (0,01) oder Tausendstelsekunden (0,001).
PV	Der Wert von PV wird in den Sollwert des Zeitgebers kopiert, wenn der Zeitgeber freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet,wenn der Istwert kleiner als der Sollwert ist. Der Zustand von Q ist nullspannungssicher, beim Einschalten erfolgt keine automatische Initialisierung.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
PV	•	•	•	•	•		•	•	•	•	•	•
Q	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

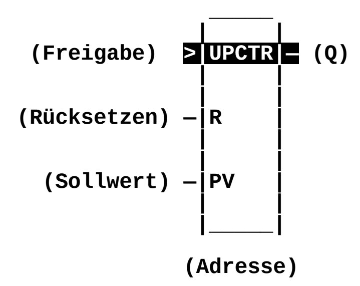
Im nachstehenden Beispiel wird mit einem OFDT-Zeitgeber ein Ausgang (%Q0001) immer dann abgeschaltet, wenn ein Eingang (%I0001) eingeschaltet wird. Der Ausgang wird 0,3 Sekunden nach Abschalten des Eingangs wieder eingeschaltet.

```
|          _____
|%I0001   |     |                                                  %Q0001
|--| |----|OFDT |--------------------------------------------------(/)-
|         |0,10s|
|         |     |
|         |     |
| CONST --|PV   |
| +00003  |_____|
|
|        %R00019
```

UPCTR

Mit der Funktion des Aufwärtszählers (UPCTR) wird bis zu einem vorgegebenen Wert vorwärts gezählt. Der Wertebereich liegt dabei zwischen 0 und +32.767 Zählwerten. Wird der Rücksetzeingang des Aufwärtszählers aktiviert, dann wird der Istwert auf Null gesetzt. Wechselt der Freigabeeingang von AUS auf EIN, wird der Istwert um 1 erhöht. Der Istwert kann über den Sollwert (PV) hinaus erhöht werden. Der Ausgang ist durchgeschaltet, solange der Istwert größer als der Sollwert oder gleich diesem ist.

Der Zustand der UPCTR-Funktion ist nullspannungssicher, beim Einschalten findet keine automatische Initialisierung statt.



Parameter

Parameter	Beschreibung
Adresse	Die UPCTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: • Istwert (CV) = Wort 1. • Sollwert (PV) = Wort 2. • Steuerwort = Wort 3. Bei der Eingabe der UPCTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. **Hinweis:** Verwenden Sie diese Adresse nicht bei anderen Aufwärts- oder Abwärtszählern oder bei anderen Anweisungen. Dies führt zu Störungen im Betrieb. **Vorsicht:** Adressüberschneidungen führen zu Störungen im Zählerbetrieb.
Freigabe	Bei einem positiven Übergang am Freigabeeingang wird der Istwert um 1 erhöht.
R	Der Istwert wird auf Null rückgesetzt, wenn R Stromfluss empfängt.
PV	Der Wert von PV wird in den Sollwert des Zählers kopiert, wenn der Zähler freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert größer als der Sollwert oder gleich diesem ist.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

- Zulässiger Datentyp oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

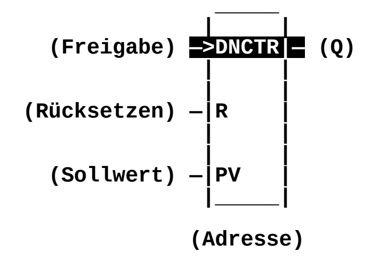
Im nachstehenden Beispiel wird der Aufwärtszähler PRT_CNT jedesmal um 1 weitergeschaltet, wenn der Eingang %I0012 von AUS auf EIN wechselt. Der interne Merker %M0001 wird durchgeschaltet, wenn 100 Einheiten gezählt wurden. Der Zählwert wird auf Null gesetzt, wenn %M0001 EIN ist.

```
|           _____
| %I0012 |      |                                              %M0001
|——| |—— >UPCTR|————————————————————————————————————————————————( )—
|        |      |
|%M0001  |      |
|——| |———|R     |
|        |      |
|        |      |
| CONST —|PV    |
| +00100 |      |
|        |      |
|        |______|
|        PRT_CNT
```

DNCTR

Die Funktion des Abwärtszählers (DNCTR) zählt von einem vorgegebenen Wert aus rückwärts. Der kleinste einstellbare Vorgabewert beträgt dabei 0, der größte +32.767. Der kleinste Istwert beträgt –32.768. Wird der Rücksetzeingang des Abwärtszählers aktiviert, dann wird der Istwert auf den Sollwert (PV) gesetzt. Wechselt der Freigabeeingang von AUS auf EIN, dann wird der Istwert um 1 erniedrigt. Der Ausgang ist durchgeschaltet, solange der Istwert negativ oder gleich Null ist.

Der Istwert der DNCTR-Funktion ist nullspannungssicher, beim Einschalten wird die Funktion nicht automatisch initialisiert.



Parameter

Parameter	Beschreibung
Adresse	Die DNCTR-Funktion benutzt drei aufeinanderfolgende Worte (Register) im %R-Speicher, um folgende Werte zu speichern: Istwert (CV) = Wort 1. Sollwert (PV) = Wort 2. Steuerwort = Wort 3. Bei der Eingabe der DNCTR-Funktion müssen Sie die Speicheradresse dieser drei aufeinanderfolgenden Worte (Register) unmittelbar unter der Funktionsgraphik angeben. **Hinweis:** Verwenden Sie diese Adresse nicht bei anderen Aufwärts- oder Abwärtszählern oder bei anderen Anweisungen. Dies führt zu Störungen im Betrieb. **Vorsicht:** Adressüberschneidungen führen zu Störungen im Zählerbetrieb.
Freigabe	Bei einem positiven Übergang am Freigabeeingang wird der Istwert um 1 erniedrigt.
R	Der Istwert wird auf den Sollwert gesetzt, wenn R Stromfluss empfängt.
PV	Der Wert von PV wird in den Sollwert des Zählers kopiert, wenn der Zähler freigegeben oder rückgesetzt wird.
Q	Der Ausgang Q wird durchgeschaltet, wenn der Istwert negativ oder gleich Null ist.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
R	•											
PV		•	•	•	•		•	•	•	•	•	•
Q	•											•

• Zulässiger Datentyp oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

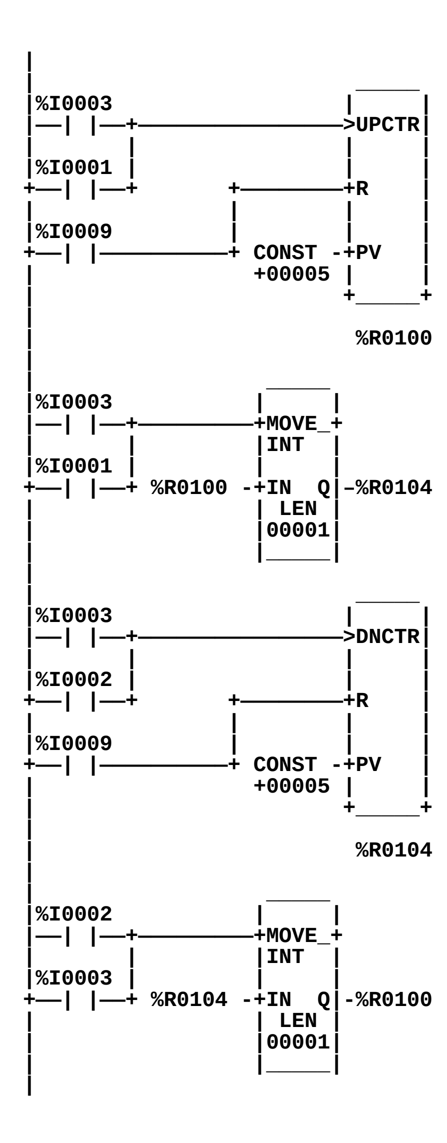
Im folgenden Beispiel zählt der mit COUNTP bezeichnete Abwärtszähler 5000 neue Teile, ehe der Ausgang %Q0005 durchgeschaltet wird.

```
|             _____
|NEW_PRT     |     |                                                  %Q0005
|——| |——    >DNCTR|——————————————————————————————————————————————————( )—
|            |     |
|NXT_BAT     |     |
|——| |———————|R    |
|            |     |
|            |     |
| CONST —————|PV   |
| +05000     |     |
|            |_____|
|             COUNTP
|
```

Beispiel

Im folgenden Beispiel wird die SPS dazu eingesetzt, die Anzahl der in einem Zwischenlager enthaltenen Teile zu verfolgen. Mit dem Befehlsvorrat der Series 90-30/20/Micro kann diese Aufgabe auf zwei verschiedene Arten gelöst werden.

Bei der ersten Methode werden ein Auf- und ein Abwärtszähler mit einem gemeinsam benutzten Register für den Istwert eingesetzt. Wird ein Teil in das Zwischenlager eingebracht, wird der Aufwärtszähler um 1 erhöht. Hierdurch erhöht sich der aktuelle Wert für die Anzahl im Zwischenlager gespeicherter Teile um 1. Verlässt ein Teil das Zwischenlager, wird der Abwärtszähler um 1 erniedrigt und verkleinert dadurch den aktuellen Wert für die Anzahl im Zwischenlager gespeicherter Teile um 1. Um einen Konflikt mit dem gemeinsam genutzten Register zu vermeiden, verwenden die beiden Zähler unterschiedliche Registeradressen. Zählt ein Register, muss sein Istwert an das aktuelle Werteregister des anderen Zählers angepasst werden.



Bei der nachstehend gezeigten zweiten Methode wird die Lagerverwaltung mit ADD- und SUB-Funktionen durchgeführt.

```
|
|
|%I0004                                                              %M0001
+——| |——————————————————————————————————————————————————————————————————(↑)—
|
|
|%I0005                                                              %M0002
+—| |———————————————————————————————————————————————————————————————————(↑)—
|
|
|          _____
|%M0001   |     |
|—| |———| ADD_|—
|         |     |
|         | INT |
|         |     |—
|%R0201 —|I1  Q|—%R00201
|         |     |
|         |     |
| CONST —|I2   |
| +00001 |_____|
|
|
|          _____
|%M0002   |     |
|—| |———| SUB_|—
|         |     |
|         | INT |
|         |     |
|%R0201 —|I1  Q|—%R00201
|         |     |
|         |     |
| CONST —|I2   |
| +00001 |_____|
|
```

Kapitel 6 Arithmetische Funktionen

In diesem Kapitel werden die arithmetischen Funktionen des Befehlssatzes der Serie 90-30/20/Micro beschrieben.

Abkürzung	Funktion	Beschreibung	Seite
ADD	Addition	Addiert zwei Zahlen.	6-2
SUB	Subtraktion	Subtrahiert eine Zahl von einer anderen.	6-2
MUL	Multiplikation	Multipliziert zwei Zahlen miteinander.	6-2
DIV	Division	Dividiert eine Zahl durch eine andere, im Ergebnis steht ein Quotient.	6-2
MOD	Modulo-Division	Dividiert eine Zahl durch eine andere, im Ergebnis steht ein Rest.	6-6
SQRT	Quadratwurzel	Zieht die Quadratwurzel aus einer ganzen oder reellen Zahl.	6-8
SIN, COS, TAN, ASIN, ACOS, ATAN	Trigonometrische Funktionen †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	6-10
LOG, LN EXP, EXPT	Logarithmus-/Exponentialfunktionen †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	6-12
RAD, DEG	Bogenmaßumwandlung †	Führt die entsprechende Funktion mit dem an Eingang IN anliegenden reellen Wert durch.	6-14

† Trigonometrische Funktionen, Logarithmus-/Exponentialfunktionen und Bogenmaß-Umwandlungsfunktionen stehen nur bei den CPU-Modellen 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352 zur Verfügung.

Hinweis

Division und Modulo-Division sind ähnliche Funktionen, die sich in ihrem Ausgangswert unterscheiden. Während die Division einen Quotienten ermittelt, ermittelt die Modulo-Division einen Rest.

Grundrechenarten (ADD, SUB, MUL, DIV)

Zu den Grundrechenarten gehören Addition, Subtraktion, Multiplikation und Division. Empfängt eine dieser Funktionen Stromfluss, wird die entsprechende arithmetische Funktion mit den Eingangsparametern I1 und I2 durchgeführt. Diese beiden Parameter müssen vom gleichen Datentyp sein. Der Ausgangswert Q ist vom gleichen Datentyp wie I1 und I2.

Hinweis

Die DIV-Funktion rundet immer ab auf die nächste ganze Zahl.
(Beispiel: 24 DIV 5 = 4.)

Die Grundrechenfunktionen benutzen die folgenden Datentypen:

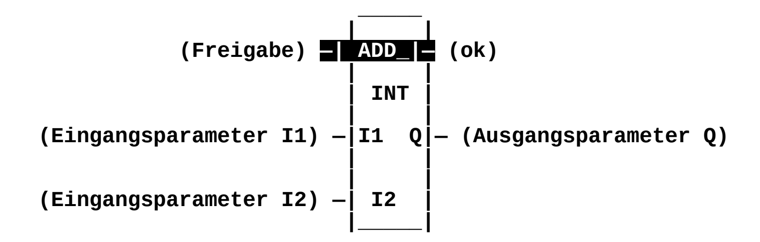
Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.
REAL	Gleitpunktzahl

Hinweis

Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Kapitel 2, Abschnitt 2 "Programmorganisation und Anwenderreferenzen/-daten” dieses Handbuchs.

Ergibt die Operation von INT oder DINT einen Überlauf, wird die Ausgangsreferenz auf den für diesen Datentyp größtmöglichen Wert gesetzt. Bei Zahlen mit Vorzeichen zeigt das Vorzeichen die Richtung des Überlaufs an. Ergibt die Operation keinen Überlauf (und liegen die Eingangswerte im zulässigen Bereich), wird der OK-Ausgang durchgeschaltet, andernfalls ist er AUS. Bei vorzeichenbehafteten oder doppeltgenauen ganzen Zahlen richtet sich das Vorzeichen des Ergebnisses der DIV- und MUL-Funktionen nach den Vorzeichen von I1 und I2.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den ersten in der Operation verwendeten Wert. (I1 steht auf der linken Seite der mathematischen Gleichung. Zum Beispiel: I1 — I2).
I2	I2 enthält eine Konstante oder Referenz für den zweiten in der Operation verwendeten Wert. (I2 steht auf der rechten Seite der mathematischen Gleichung. Zum Beispiel: I1 — I2).
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und keine unzulässige Operation aufgetreten ist.
Q	Der Ausgang Q enthält das Ergebnis der Operation.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
OK	•											•
Q		o	o	o	o		o	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
† Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

Hinweis

INT ist der Standardtyp für 16-Bit- oder Einzelregister-Operanden. Drücken Sie **F10**, um die Typeneinstellung auf DINT, 32-Bit-Doppelwort, oder REAL (nur CPUs 35x und 36x) umzuschalten. Die INT-Werte der SPS belegen ein einzelnes 16-Bit-Register, %R, %AI oder %AQ. Die DINT-Werte hingegen benötigen zwei aufeinanderfolgende Register, wobei die niedrigstwertigen 16 Bits im ersten Wort und die höchstwertigen 16 Bits im zweiten Wort stehen. REAL-Werte (CPUs 35x und 36x, Ausgabestand 9 und höher, sowie alle Ausgabestände der CPU352) belegen ebenfalls ein 32-Bit-Doppelregister, wobei das Vorzeichen im höchstwertigen Bit von Exponent und Mantisse gefolgt wird.

Beispiel

In dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird und die Subtraktion ohne Überlauf ausgeführt wurde, der ganzzahlige Inhalt von %R0002 um 1 erniedrigt und die Spule %Q0001 durchgeschaltet.

```
|           _____
|%I0001    |     |                                                %Q0001
|—| |——————| SUB_|————————————————————————————————————————————————( )—
|          |     |
|          | INT |
|          |     |
|%R0002 —  |I1  Q|—%R0002
| +00095   |     |
|          |     |
| CONST —  |I2   |
| +00001   |_____|
|
```

Arithmetische Funktionen und Datentypen

Funktion	Operation	Anzeige als
ADD INT	Q (16 Bits) = I1 (16 Bits) + I2 (16 Bits)	5-stellige Dezimalzahl mit Vorzeichen
ADD DINT	Q (32 Bits) = I1 (32 Bits) + I2 (32 Bits)	8-stellige Dezimalzahl mit Vorzeichen
ADD REAL*	Q (32 Bits) = I1 (32 Bits) + I2 (32 Bits)	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
SUB INT	Q (16 Bits) = I1 (16 Bits) – I2 (16 Bits)	5-stellige Dezimalzahl mit Vorzeichen
SUB DINT	Q (32 Bits) = I1 (32 Bits) – I2 (32 Bits)	8-stellige Dezimalzahl mit Vorzeichen
SUB REAL*	Q (32 Bits) = I1 (32 Bits) – I2 (32 Bits)	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
MUL INT	Q (16 Bits) = I1 (16 Bits) * I2 (16 Bits)	5-stellige Dezimalzahl mit Vorzeichen
MUL DINT	Q (32 Bits) = I1 (32 Bits) * I2 (32 Bits)	8-stellige Dezimalzahl mit Vorzeichen
MUL REAL*	Q (32 Bits) = I1 (32 Bits) * I2 (32 Bits)	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen
DIV INT	Q (16 Bits) = I1 (16 Bits) / I2 (16 Bits)	5-stellige Dezimalzahl mit Vorzeichen
DIV DINT	Q (32 Bits) = I1 (32 Bits) / I2 (32 Bits)	8-stellige Dezimalzahl mit Vorzeichen
DIV REAL*	Q (32 Bits) = I1 (32 Bits) / I2 (32 Bits)	7-stellige Dezimalzahl mit Vorzeichen und Nachkommastellen

* Nur CPUs der Serien 35x und 36x, Ausgabestand 9 oder höher, sowie alle Ausgabestände der CPU352

Hinweis

Die Datentypen der Eingangs- und Ausgangsdaten müssen gleich sein. Im Gegensatz zur Series 90-70 sind hier bei den Funktionen MUL und DIV keine gemischten Typen möglich. Beispiel: MUL INT von zwei 16-Bit-Eingangswerten ergibt ein 16-Bit-Produkt, keinen 32-Bit-Wert. Soll bei MUL DINT ein 32-Bit-Ergebnis herauskommen, müssen beide Eingangswerte 32-Bit-Werte sein. Bei DIV INT liefert ein 16-Bit-I2 ein 16-Bit-Ergebnis, während DIV DINT ein 32-Bit-I1 durch ein 32-Bit-I2 teilt und ein 32-Bit-Ergebnis liefert.

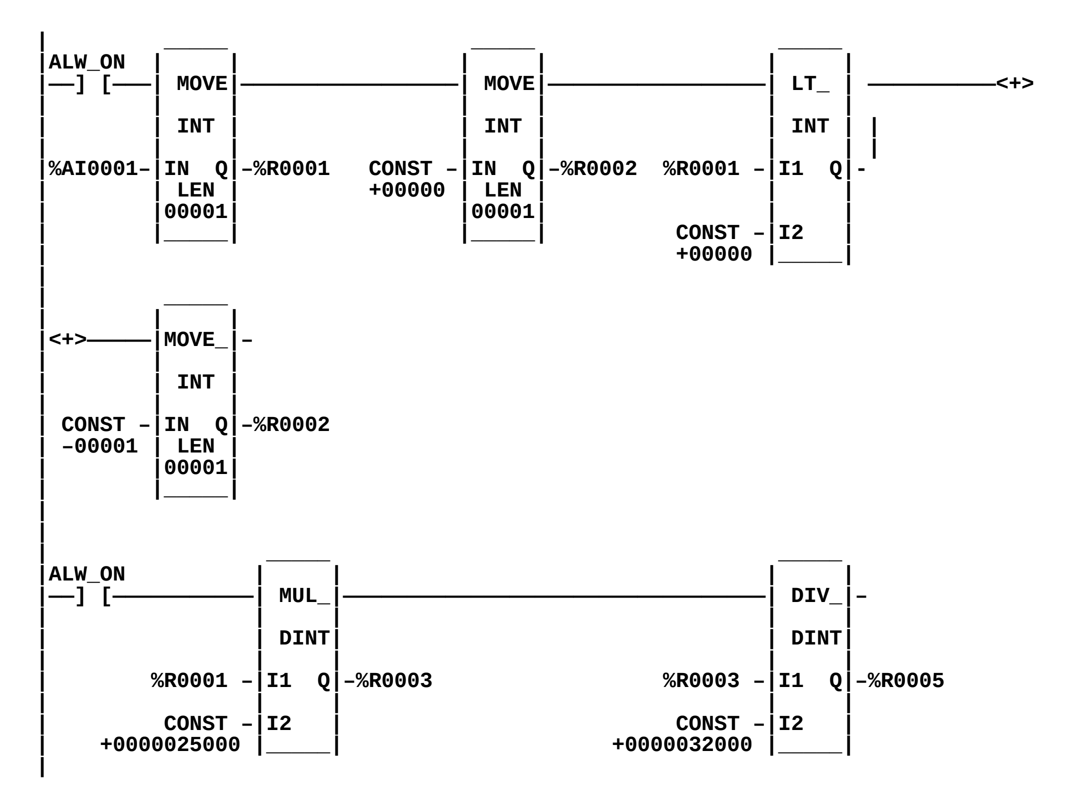
Diese Funktionen schalten den Stromfluss durch, wenn kein arithmetischer Überlauf entsteht. Bei einem Überlauf ist das Ergebnis der größte Wert mit richtigem Vorzeichen, und es findet kein Stromfluss statt.

Vermeiden Sie Überläufe beim Einsatz der MUL- und DIV-Funktionen. Beachten Sie beim Konvertieren von INT- Werten in DINT-Werte, dass die CPU Standard-Zweierkomplemente verwendet, bei denen das Vorzeichen im höchsten Bit des zweiten Wortes steht. Sie müssen das Vorzeichen des unteren 16-Bit-Wortes prüfen und es in das zweite 16-Bit-Wort übertragen. Ist das höchstwertige Bit in einem 16-Bit-INT-Wort 0 (positiv), müssen Sie eine Null in das zweite Wort schieben. Ist das höchstwertige Bit in einem 16-Bit-Wort –1 (negativ), müssen Sie –1 (Hex 0FFFFh) in das zweite Wort schieben. Die Konvertierung von DINT nach INT ist einfacher, da das niedrigwertige 16-Bit-Wort (erstes Register) den INT-Teil des 32-Bit-DINT-Wortes darstellt. Die oberen 16 Bits oder das zweite Wort müssen entweder 0 (positiv) oder –1 (negativ) sein, andernfalls ist die DINT-Zahl zu groß, um auf 16 Bits konvertiert zu werden.

Beispiel

Eine häufige Anwendung ist die Skalierung analoger Eingangswerte mit einer MUL-Operation, die von einer DIV- und möglicherweise einer ADD-Operation gefolgt wird. Bei einem Bereich bis 32000 ergibt die Verwendung von MUL INT einen Überlauf. Die Verwendung eines %AI-Wertes für MUL DINT funktioniert ebenfalls nicht, da der 32-Bit-Eingangswert I1 gleichzeitig zwei Analogeingänge kombiniert. Sie müssen den Analogeingang in das untere Wort eines Doppelregisters kopieren, dann das Vorzeichen prüfen und daraufhin das zweite Register auf 0 (wenn positiv) oder –1 (wenn negativ) setzen. Verwenden Sie das Doppelregister mit MUL DINT um ein 32-Bit-Produkt für die folgende DIV-Funktion zu erzielen.

Mit den folgenden Programmbeispiel können Sie einen +/–10-V-Eingang %AI auf +/–25000 technische Einheiten in %R5 skalieren.



MOD (INT, DINT)

Mit der Modulofunktion wird ein Wert durch einen anderen Wert vom gleichen Datentyp dividiert. Das Ergebnis ist ein Rest. Das Vorzeichen des Ergebnisses ist immer gleich dem Vorzeichen des Eingangsparameters I1.

Die MOD-Funktion verarbeitet die folgenden Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.

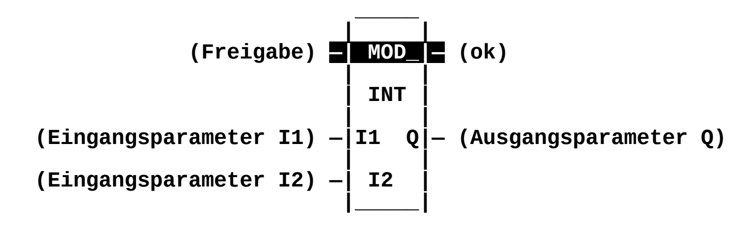
Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Kapitel 2, Abschnitt 2 "Programmorganisation und Anwenderreferenzen/-daten" dieses Handbuchs.

Wenn die Modulofunktion Stromfluss erhält, dividiert sie den Wert I1 durch dem Wert I2. Beide Werte müssen vom gleichen Datentyp sein. Der Ausgangswert Q berechnet sich zu:

Q = I1 - ((I1 DIV I2) * I2)

DIV ergibt dabei eine ganze Zahl. Q ist vom gleichen Datentyp wie die Eingangsparameter I1 und I2.

Der OK-Ausgang ist immer EIN, wenn die Funktion Stromfluss erhält und nicht versucht wurde, durch Null zu dividieren. In einem solchen Fall wird der OK-Ausgang auf AUS gesetzt.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den Dividenden.
I2	I2 enthält eine Konstante oder Referenz für den Divisor.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde.
Q	Der Ausgang Q enthält den Rest der Division von I1 durch I2.

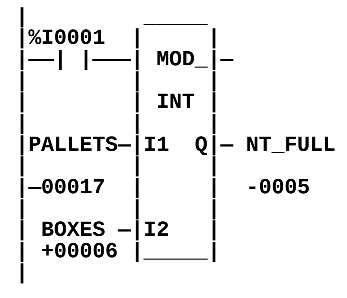
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
OK	•											•
Q		o	o	o	o		o	•	•	•		

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT.
- † Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

Beispiel

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, der Rest der ganzzahligen Division von PALLETS durch BOXES in NT_FULL eingetragen.



SQRT (INT, DINT, REAL)

Mit der Quadratwurzelfunktion (SQRT) können Sie die Quadratwurzel aus einem Wert ziehen. Empfängt die Funktion Stromfluss, wird der Wert von Ausgang Q auf den ganzzahligen Teil der Quadratwurzel vom Eingang IN gesetzt. Die Datentypen von Ausgang Q und Eingang IN müssen gleich sein.

Die SQRT-Funktion verarbeitet die folgenden Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.
REAL	Gleitpunktzahl

Hinweis

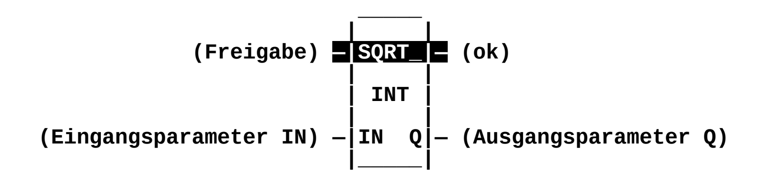
Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Kapitel 2, Abschnitt 2 "Programmorganisation und Anwenderreferenzen/-daten” dieses Handbuchs.

Der Ausgang OK wird durchgeschaltet, wenn die Funktion ohne Überlauf durchgeführt wurde und keine der folgenden unzulässigen REAL-Operationen auftrat:

- IN < 0.
- IN ist keine Zahl

In den anderen Fällen wird OK abgeschaltet.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Konstante oder Referenz für den Wert, dessen Quadratwurzel berechnet werden soll. Ist IN kleiner als Null, wird die Funktion nicht durchgeschaltet.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und keine unzulässige Operation aufgetreten ist.
Q	Ausgang Q enthält die Quadratwurzel von IN.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•†	
OK	•											•
Q		o	o	o	o		o	•	•	•		

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
- † Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf Werte zwischen -32.768 und +32.767 beschränkt.

Beispiel

Im nachstehenden Beispiel wird die Quadratwurzel aus der in in %AI001 stehenden ganzen Zahl in das in %R0003 stehende Ergebnis eingetragen, wenn %I0001 EIN ist.

```
|            _____
|%I0001     |     |
|——| |——————|SQRT_|
|           |     |
|           | INT |
|           |     |
|%AI0001————|IN  Q|—%R0003
|           |_____|
|
```

Trigonometrische Funktionen (SIN, COS, TAN, ASIN, ACOS, ATAN)

Die Funktionen SIN, COS und TAN ermitteln den Sinus, Cosinus und Tangens des an ihrem Eingang anliegenden Wertes. Empfängt eine der Funktionen Stromfluss, berechnet sie den entsprechenden trigonometrischen Wert (Sinus, Cosinus oder Tangens) von IN (in Radiant) und speichert das Ergebnis im Ausgang Q. IN und Q sind beide Gleitpunktwerte.

Die Funktionen ASIN, ACOS und ATAN ermitteln die Umkehrfunktionen des Sinus, Cosinus und Tangens des an ihrem Eingang anliegenden Wertes. Empfängt eine der Funktionen Stromfluss, berechnet sie den entsprechenden trigonometrischen Wert (Arcussinus, Arcuscosinus oder Arcustangens) von IN (in Radiant) und speichert das Ergebnis im Ausgang Q. IN und Q sind beide Gleitpunktwerte.

Bei den Funktionen SIN, COS und TAN ist ein weiter Eingangsbereich möglich:
$-2^{63} < \text{IN} < +2^{63}$, ($2^{63} \approx 9{,}22 \times 10^{18}$).

Der Eingangsbereich der Funktionen ASIN und ACOS ist schmal: $-1 \leq \text{IN} \leq 1$. Hat der Parameter IN einen gültigen Wert, liegt das Ergebnis Q der Funktion ASIN_REAL im Bereich:

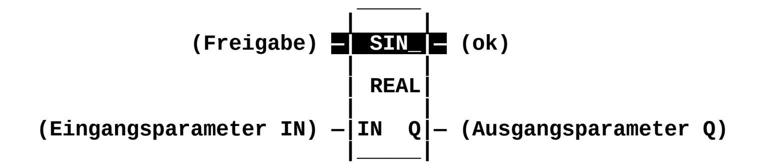
$$\textbf{ASIN (IN)} = -\frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$

Das Ergebnis Q der Funktion ACOS_REAL liegt im Bereich:

$$\textbf{ACOS (IN)} = 0 \leq Q \leq \pi$$

Bei der Funktion ATAN ist der größte Eingangsbereich möglich: $-\infty \leq \text{IN} \leq +\infty$. Hat der Parameter IN einen gültigen Wert, liegt das Ergebnis Q der Funktion ATAN_REAL im Bereich:

$$\textbf{ATAN (IN)} = -\frac{\pi}{2} \leq Q \leq \frac{\pi}{2}$$



Hinweis

Die trigonometrischen Funktionen gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Konstante oder eine Referenz auf den reellen Wert für die Operation.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde, keine unzulässige Operation aufgetreten ist und IN eine Zahl ist.
Q	Ausgang Q enthält den trigonometrischen Wert von IN.

Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN								•	•	•	•	
OK	•											•
Q								•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Im nachstehenden Beispiel wird der Cosinus des Werte in %R0001 in %R0033 eingetragen.

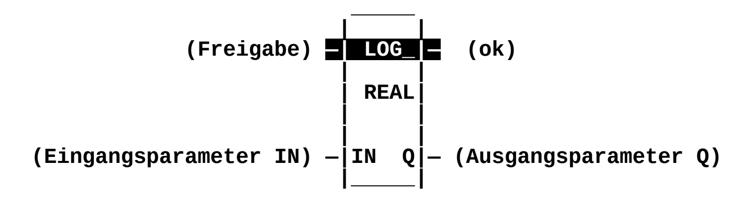
```
|             _____
|ALW_ON      |     |
|——] [———————| COS_|—
|            |     |
|            | REAL|
|            |     |
|     %R0001—|IN  Q|—%R0033
|            |     |
|   +3.141500|_____| -1.000000
|
```

Logarithmus- / Exponentialfunktionen (LOG, LN, EXP, EXPT)

Die Funktionen LOG, LN und EXP besitzen zwei Eingangsparameter und zwei Ausgangsparameter. Wenn diese Funktionen Stromfluss empfangen, führen sie mit dem am Eingang IN anliegenden reellen Wert die entsprechende Logarithmus- oder Exponentialfunktion aus und legen das Ergebnis im Ausgang Q ab.

- Bei der Funktion LOG wird der dekadische Logarithmus von IN in Q abgelegt.
- Bei der Funktion LN wird der natürliche Logarithmus von IN in Q abgelegt.
- Bei der Funktion EXP wird ***e*** zu der an IN angegebenen Potenz erhoben und das Ergebnis in Q abgelegt.
- Bei der Funktion EXPT wird der Wert am Eingang I1 zu der an I2 angegebenen Potenz erhoben und das Ergebnis in Q abgelegt (die Funktion EXPT besitzt drei Eingangs- und zwei Ausgangsparameter).

Der Ausgang OK wird durchgeschaltet, sofern IN eine Zahl und nicht negativ ist.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den reellen Wert für die Operation.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde, keine unzulässige Operation aufgetreten ist und IN eine Zahl und nicht negativ ist.
Q	Ausgang Q enthält den Logarithmus- bzw. Exponentialwert von IN.

Hinweis

Die Funktionen LOG, LN, EXP und EXPT stehen nur bei den CPU-Modellen 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352 zur Verfügung.

Hinweis

Ist der Eingangswert IN bei der Funktion EXP minus unendlich (-∞), gibt die Funktion erwartungsgemäß Null aus. Bei der CPU352 schaltet die Funktion den Stromfluss *nicht* durch. Bei allen anderen CPUs 90-30 CPUs *schaltet die Funktion den Stromfluss durch*, selbst wenn der Ausgangswert Null ist.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN*								•	•	•	•	
OK	•											•
Q								•	•	•		

* Bei der Funktion EXPT wird Eingang IN durch die Eingangsparameter I1 und I2 ersetzt.

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

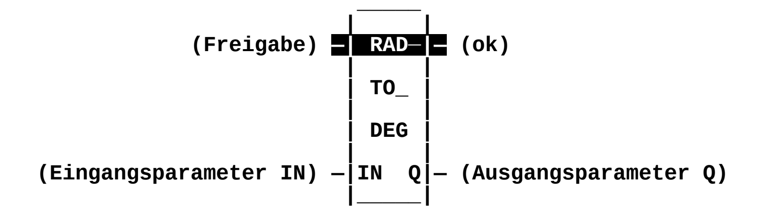
Im nachstehenden Beispiel wird der Wert von %AI001 zur 2,5-ten Potenz erhoben. Das Ergebnis wird in%R0001 eingetragen.

```
|               _____
|ALW_ON        |     |
|——] [————————|EXPT_|—
|              |     |
|              | REAL|
|              |     |
|              |     |
|      %AI0001—|I1  Q|—%R0001
|              |     |
|              |     |
|       CONST —|I2   |
|   2.50000E+00|_____|
|
```

Bogenmaß-Umrechnung (RAD, DEG)

Empfängt die Funktion Stromfluss, wird die entsprechende Konvertierung (RAD_TO_DEG = Bogenmaß in Grad oder DEG_TO_RAD = Grad in Bogenmaß) des reellen Wertes am Eingang IN durchgeführt. Das Ergebnis wird in den Ausgang Q eingetragen.

Der Ausgang OK wird durchgeschaltet, sofern IN eine Zahl ist.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den reellen Wert für die Operation.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion ohne Überlauf ausgeführt wurde und IN eine Zahl ist.
Q	Ausgang Q enthält den konvertierten Wert von IN.

Hinweis

Die Bogenmaß-Umrechnungsfunktionen gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

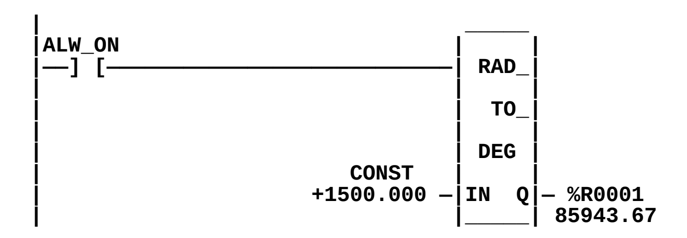
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN								•	•	•	•	
OK	•											•
Q								•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Im nachstehenden Beispiel wird +1500 in Grad umgewandelt und in %R0001 eingetragen.



Kapitel 7

Relationale Funktionen

Mit relationalen Funktionen werden zwei Zahlen miteinander verglichen. In diesem Kapitel werden folgende relationale Funktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
EQ	Gleich	Prüft, ob zwei Zahlen gleich sind.	7-2
NE	Ungleich	Prüft, ob zwei Zahlen ungleich sind.	7-2
GT	Größer als	Prüft, ob eine Zahl größer als eine andere ist.	7-2
GE	Größer als oder gleich	Prüft, ob eine Zahl größer als eine andere oder mit dieser gleich ist.	7-2
LT	Kleiner als	Prüft, ob eine Zahl kleiner als eine andere ist.	7-2
LE	Kleiner als oder gleich	Prüft, ob eine Zahl kleiner als eine andere oder mit dieser gleich ist.	7-2
RANGE	Bereich	Stellt fest, ob eine Zahl in einem bestimmten Bereich liegt (verfügbar für CPUs ab Ausgabestand 4.5)	7-4

Relationale Standardfunktionen (EQ, NE, GT, GE, LT, LE)

Empfängt die Funktion Stromfluss, vergleicht sie Eingangsparameter I1 mit Eingangsparameter I2. Der Datentyp beider Parameter muss gleich sein. Relationale Funktionen verwenden folgende Datentypen:

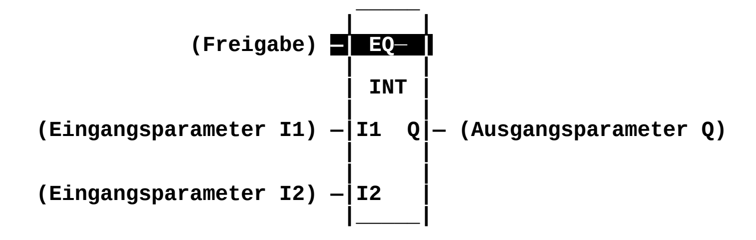
Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.
REAL	Gleitpunktzahl

Hinweis

Der Datentyp REAL ist nur in den CPUs 35x und 36x ab Ausgabestand 9 sowie bei allen Ausgabeständen der CPU352 verfügbar. Das Bit %S0020 wird gesetzt, wenn eine relationale Funktion mit REAL-Daten erfolgreich ausgeführt wird. Das Bit wird gelöscht, wenn einer der Eingänge keine Zahl ist. Der Funktionsblock RANGE akzeptiert den Datentyp REAL nicht.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der relationalen Funktion in ganze Zahl mit Vorzeichen, doppeltgenaue ganze Zahl mit Vorzeichen, oder reelle Zahl verändert werden. Wollen Sie Daten eines anderen Typs oder zwei unterschiedliche Datentypen miteinander vergleichen, müssen Sie zunächst die entsprechende Konvertierungsfunktion (siehe Kapitel 11 "Konvertierungsfunktionen") anwenden, um die Daten in einen der unterstützten Datentypen umzuwandeln.

Erfüllen die Eingangsparameter I1 und I2 die angegebene Relation, empfängt Ausgang Q Stromfluss und wird durchgeschaltet (1); andernfalls wird er auf AUS (0) gesetzt.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für den ersten zu vergleichenden Wert. (I1 steht auf der linken Seite der relationalen Gleichung; zum Beispiel I1 < I2).
I2	I2 enthält eine Konstante oder Referenz für den zweiten zu vergleichenden Wert. (I2 steht auf der rechten Seite der relationalen Gleichung; zum Beispiel I1 < I2).
Q	Der Ausgang Q wird durchgeschaltet, wenn I1 und I2 der angegebenen Relation entsprechen.

Hinweis

I1 und I2 müssen gültige Zahlen sein.

Erweiterte Beschreibung

Funktion	Beschreibung
Gleich	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 gleich dem Wert an Eingang I2 ist.
Ungleich	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 NICHT gleich dem Wert an Eingang I2 ist.
Größer als	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 größer als der Wert an Eingang I2 ist.
Größer als oder gleich	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 größer als der Wert an Eingang I2 oder mit diesem gleich ist.
Kleiner als	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 kleiner als der Wert an Eingang I2 ist.
Kleiner als oder gleich	Ausgang Q wird durchgeschaltet, wenn die Funktion freigegeben und der Wert an Eingang I1 kleiner als der Wert an Eingang I2 oder mit diesem gleich ist.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		o	o	o	o		o	•	•	•	•†	
I2		o	o	o	o		o	•	•	•	•†	
Q	•											•

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT-Daten zulässig, nicht für DINT oder REAL.
- † Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf ganzzahlige Werte beschränkt.

Beispiel

Bei dem folgenden Beispiel werden jedesmal, wenn der Eingang %I0001 gesetzt wird, die beiden doppeltgenauen ganzzahligen Werte mit Vorzeichen PWR_MDE und BIN_FUL miteinander verglichen. Die Spule %Q0002 wird eingeschaltet, wenn PWR_MDE kleiner als BIN_FUL oder mit dessem Wert gleich ist.

```
|                 _____
|%I0001          |     |
|——| |———————————| LE_ |
|                |     |
|                | DINT|
|                |     |                                              %Q0002
|       PWR_MDE—|I1  Q|———————————————————————————————————————————————( )—
|                |     |
|                |     |
|       BIN_FUL—|I2   |
|                |_____|
|
```

RANGE (INT, DINT, WORD)

Mit der RANGE-Funktion können Sie feststellen, ob ein Wert in einem von zwei Zahlen festgelegten Bereich liegt.

Hinweis

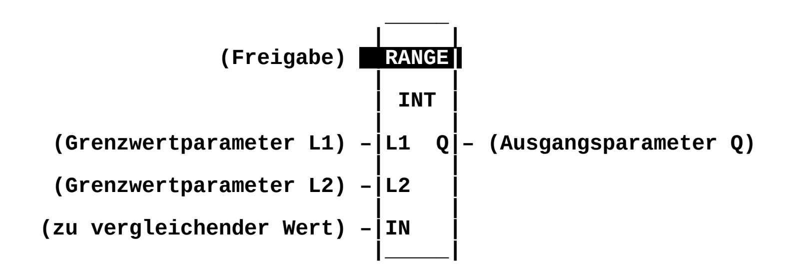
Diese Funktion ist *nur* verfügbar in CPUs ab Ausgabestand 4.41.

Die RANGE-Funktion verarbeitet die folgenden Datentypen:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.
WORD	Wort.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Weitere Informationen über Datentypen finden Sie in Kapitel 2, Abschnitt 2 "Programmorganisation und Anwenderreferenzen/-daten" dieses Handbuchs.

Ist die Funktion freigegeben, vergleicht der RANGE-Funktionsblock den Wert im Eingangsparameter IN mit dem durch die Grenzwertparameter L1 und L2 festgelegten Bereich. Liegt der Wert in diesem Bereich (einschließlich der Grenzwerte), wird Ausgangsparameter Q durchgeschaltet (1). Im anderen Fall ist Q AUS (0).



Hinweis

Die Grenzparameter L1 und L2 stellen die Endpunkte des Bereichs dar. Keiner dieser Parameter hat eine Bedeutung als Minimal-/Maximalwert oder oberer/unterer Grenzwert. Ein gewünschter Bereich von 0 bis 100 kann daher definiert werden mit L1 = 0 und L2 = 100 oder aber mit L1 =100 und L2 = 0.

Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
L1	L1 enthält den Anfangspunkt des Bereichs.
L2	L2 enthält den Endpunkt des Bereichs.
IN	IN enthält den Wert, der mit dem durch L1 und L2 festgelegten Bereich verglichen wird.
Q	Ausgang Q wird aktiviert, wenn der Wert aus IN innerhalb dem durch L1 und L2 festgelegten Bereich liegt (einschließlich der Grenzwerte).

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
L1		o	o	o	o		o	•	•	•	•‡	
L2		o	o	o	o		o	•	•	•	•‡	
IN		o	o	o	o		o	•	•	•		
Q	•											•

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Referenz nur für INT- oder WORD-Daten zulässig, nicht für DINT.
- ‡ Bei doppeltgenauen ganzzahligen Operationen mit Vorzeichen sind Konstanten auf ganzzahlige Werte beschränkt.

Beispiel 1

Im nachstehenden Beispiel wird überprüft, ob %AI0001 in dem durch die beiden Konstanten 0 und 100 festgelegten Zahlenbereich liegt.

```
|          _____
|%I0001   |     |
|——| |————+RANGE|
|         | INT |
|         |     |                                                             %Q0002
|   100 ——+L1  Q+—————————————————————————————————————————————————————————————( )—
|         |     |
|     0 ——+L2   |
|         |     |
|%AI0001——+IN   |
|         |_____|
|
```

RANGE-Wahrheitswerttabelle				
Freigabezu-stand %I0001	Wert von L1 Konstante	Wert von L2 Konstante	Wert von IN %AI0001	Zustand von Q %Q0001
EIN	100	0	< 0	AUS
EIN	100	0	0 — 100	EIN
EIN	100	0	> 100	AUS
AUS	100	0	Nicht zutreffend	AUS

Beispiel 2

In diesem Beispiel wird untersucht, ob %AI001 in dem durch zwei Registerwerte definierten Bereich liegt.

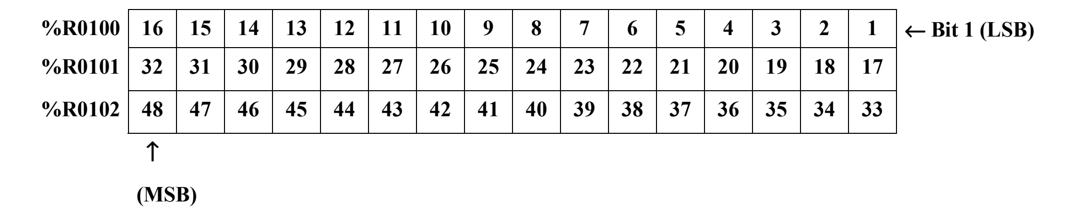
```
|          _____
|%I0001   |     |
|——| |————+RANGE|
|         | INT |
|         |     |                                                    %Q0002
|%R0001 —+L1   Q|————————————————————————————————————————————————————( )—
|         |     |
|%R0002 —+L2    |
|         |     |
|%AI0001—+IN    |
|         |_____|
|
```

RANGE-Wahrheitswerttabelle				
Freigabezustand %I0001	**Wert von L1 %R0001**	**Wert von L2 %R0002**	**Wert von IN %AI0001**	**Zustand von Q %Q0001**
EIN	500	0	< 0	AUS
EIN	500	0	0— 500	EIN
EIN	500	0	> 500	AUS
AUS	500	0	Nicht zutreffend	AUS

Kapitel 8 Bitoperationsfunktionen

Bitoperationsfunktionen führen vergleichende, logische und verschiebende Operationen mit Bitfolgen durch. Die Funktionen AND, OR, XOR und NOT bearbeiten ein einzelnes Wort. Die restlichen Bitoperationsfunktionen können mehrere Worte bearbeiten, wobei die maximale Länge einer Folge 256 Worte beträgt. Sämtliche Bitoperationsfunktionen benötigen Daten vom Typ WORD.

Obwohl die Daten im 16-Bit-Raster angegeben werden müssen, bearbeiten diese Funktionen die Daten als fortlaufendes Bitmuster, wobei Bit 1 des ersten Wortes das niedrigstwertige Bit (LSB) darstellt. Das letzte Bit des letzten Wortes ist das höchstwertige Bit (MSB). Geben Sie zum Beispiel drei Datenworte ab der Referenz %R0100 an, dann werden diese als 48 zusammenhängende Bits behandelt.



%R0100	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**← Bit 1 (LSB)**
%R0101	**32**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	
%R0102	**48**	**47**	**46**	**45**	**44**	**43**	**42**	**41**	**40**	**39**	**38**	**37**	**36**	**35**	**34**	**33**	
	↑ (MSB)																

Hinweis

Überlappende Adressbereiche bei Ein- und Ausgangsreferenzen in Mehrwortfunktionen können zu unvorhersehbaren Ergebnissen führen.

In diesem Abschnitt werden folgende Bitoperationsfunktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
AND	Logisch UND	Sind ein Bit in Bitfolge I1 und das entsprechende Bit in Bitfolge I2 beide 1, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	8-3
OR	Logisch ODER	Sind ein Bit in Bitfolge I1 und/oder das entsprechende Bit in Bitfolge I2 beide 1, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	8-3
XOR	Logisches Exklusiv-ODER	Sind ein Bit in Bitfolge I1 und das entsprechende Bit in Bitfolge I2 voneinander verschieden, wird in der gleichen Bitposition der Ausgangsbitfolge Q eine 1 eingetragen.	8-5
NOT	Logische Invertierung	Die einzelnen Bits in der Ausgangsbitfolge Q werden auf den entgegengesetzten Zustand der entsprechenden Bits in der Bitfolge I1 gesetzt.	8-7
SHL	Nach links schieben	Alle Bits in einem Wort oder einer Wortfolge werden um die angegebene Anzahl Plätze nach links verschoben.	8-8
SHR	Nach rechts schieben	Alle Bits in einem Wort oder einer Wortfolge werden um die angegebene Anzahl Plätze nach rechts verschoben.	8-8
ROL	Nach links rotieren	Alle Bits in einer Folge werden um die angegebene Anzahl Plätze nach links rotiert.	8-10
ROR	Nach rechts rotieren	Alle Bits in einer Folge werden um die angegebene Anzahl Plätze nach rechts rotiert.	8-57
BTST	Bit testen	Untersucht ein Bit in einer Bitfolge darauf, ob es momentan 1 oder 0 ist.	8-12
BSET	Bit auf 1 setzen	Setzt ein Bit in einer Bitfolge auf 1.	8-14
BCLR	Bit auf 0 setzen	Setzt ein Bit in einer Bitfolge auf 0.	8-14
BPOS	Bitposition	Findet ein auf 1 gesetztes Bit in einer Bitfolge.	8-16
MSKCMP	Vergleich mit Maske	Vergleicht den Inhalt zweier getrennter Bitfolgen mit der Möglichkeit, Bits zu maskieren (verfügbar für CPUs ab Ausgabestand 4.5).	8-18

AND und OR (WORD)

Beginnend mit dem jeweils niedrigstwertigen Bit (LSB) untersuchen die Funktionen AND und OR bei jedem Zyklus, bei dem sie Stromfluss erhalten, die einzelnen Bits in den Bitfolgen I1 und I2.

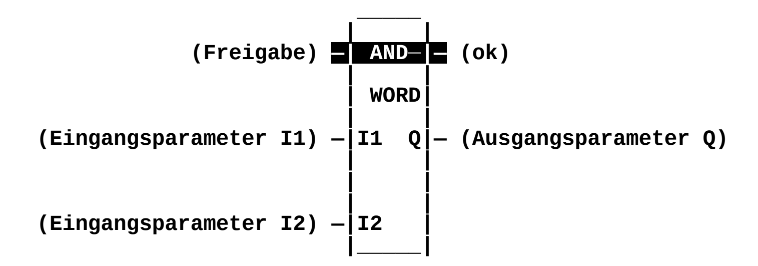
Sind zwei einander entsprechende Bits beide "1", dann trägt die AND-Funktion an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" ein. Ist auch nur eines der beiden Bits "0", dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "0" eingetragen.

Die AND-Funktion ist bei der Erstellung von Masken oder Menüs hilfreich, bei denen nur bestimmte Bits durchgelassen werden (die einer "1" in der Maske entsprechen) und alle anderen Bits auf "0" gesetzt werden. Mit dieser Funktion kann auch ein bestimmter Bereich im Wortspeicher gelöscht werden, indem die Bits mit einer anderen Bitfolge, die nur Nullen enthält, über die AND-Funktion verknüpft werden. Die angegebenen Bitfolgen I1 und I2 können sich überlappen.

Ist eines der beiden Bits "1", dann trägt die OR-Funktion an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" ein. Sind beide Bits "0", wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "0" eingetragen.

Die OR-Funktion ist hilfreich bei der Kombination von Bitfolgen und bei der Steuerung mehrerer Ausgänge durch eine einfache logische Struktur. Die Funktion entspricht der Parallelschaltung zweier Relaiskontakte für jedes in der Bitfolge enthaltene Bit. Mit ihr können Anzeigelampen aus Eingangszuständen heraus angesteuert oder Statuslampen zum Blinken gebracht werden.

Die Funktion schaltet den Stromfluss jedesmal nach rechts durch, wenn sie Stromfluss erhält.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das erste Wort der ersten Folge.
I2	I2 enthält eine Konstante oder Referenz für das erste Wort der zweiten Folge.
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
Q	Der Ausgang Q enthält das Ergebnis der Operation.

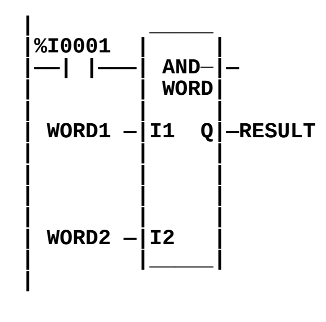
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
OK	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
† Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

Bei dem folgenden Beispiel werden jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolischen Adressen WORD1 und WORD2 spezifizierten 16 Bit langen Bitfolgen untersucht. Das Ergebnis der logischen AND-Verknüpfung wird in der Ausgangs-Bitfolge RESULT eingetragen.



WORD1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
WORD2	1	1	0	1	1	1	0	0	0	0	0	0	1	1	1	1
RESULT	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

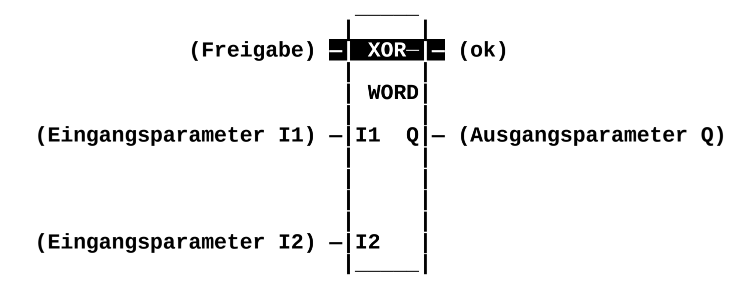
XOR (WORD)

Die XOR-Funktion (Antivalenz) vergleicht die einzelnen einander entsprechenden Bits in den Bitfolgen I1 und I2 miteinander. Unterscheiden sich diese beiden Bits, dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" eingetragen.

Beginnend mit dem jeweils niedrigstwertigen Bit (LSB) untersucht die XOR-Funktion bei jedem Zyklus, bei dem sie Stromfluss erhält, die einzelnen einander entsprechenden Bits in den Bitfolgen I1 und I2. Ist nur eines der beiden Bits "1", dann wird an der zugehörigen Stelle in der Ausgangsbitfolge Q eine "1" eingetragen. Die XOR-Funktion schaltet den Stromfluss jedesmal dann nach rechts durch, wenn sie Stromfluss erhält.

Beginnen die Bitfolge I2 und die Ausgangsbitfolge Q mit der gleichen Referenz, dann wird bei einer "1" in der Bitfolge I1 bei jedem Zyklus, bei dem die Funktion Stromfluss erhält, das entsprechende Bit in der Bitfolge I2 zwischen "0" und "1" umgeschaltet. Es können längere Zyklen programmiert werden, indem der Stromfluss zur Funktion mit der doppelten gewünschten Blinkfrequenz gepulst wird. Der Stromflussimpuls sollte mindestens einen Zyklus lang sein (Wischrelais oder automatisch rücksetzender Timer).

Mit der XOR-Funktion kann ein schneller Vergleich zweier Bitfolgen durchgeführt werden oder es kann eine Bitgruppe mit einer Frequenz von einem EIN-Zustand pro zwei Zyklen zum Blinken angeregt werden.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das erste Wort, das über die XOR-Funktion verknüpft wird.
I2	I2 enthält eine Konstante oder Referenz für das zweite Wort, das über die XOR-Funktion verknüpft wird.
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
Q	Der Ausgang Q enthält das Ergebnis der logischen XOR-Verknüpfung von I1 und I2.

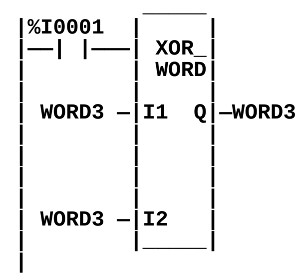
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
I2		•	•	•	•	•	•	•	•	•	•	
OK	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
† Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolische Adresse WORD3 dargestellte Bitfolge gelöscht (auf Null gesetzt).



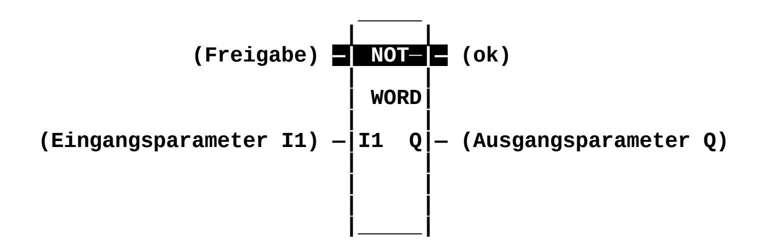
I1 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0
I2 (WORD3)	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	0

Q (WORD3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOT (WORD)

Die NOT-Funktion trägt in die einzelnen Stellen der Ausgangs-Bitfolge Q den invertierten Wert der Bits der Eingangsbitfolge I1 ein.

Bei jedem Zyklus, bei dem die Funktion Stromfluss erhält, werden alle Bits verändert, so dass die Ausgangs-Bitfolge Q ein gespiegeltes Abbild der Bitfolge I1 ist. Die NOT-Funktion schaltet den Stromfluss jedesmal dann nach rechts durch, wenn sie Stromfluss erhält.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
I1	I1 enthält eine Konstante oder Referenz für das Wort, das invertiert werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
Q	Der Ausgang Q enthält die Negation von I1.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		•	•	•	•	•	•	•	•	•	•	
OK	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann. † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, die durch die symbolische Adresse TAC dargestellte Bitfolge in die invertierte Bitfolge CAT umgewandelt.

```
|         _____
|%I0001  |     |
|--| |---| NOT-|-
|        | WORD|
|        |     |
|   CAT -|I1  Q|-TAC
|        |     |
|        |     |
```

SHL und SHR (WORD)

Die SHL-Funktion verschiebt alle Bits eines Wortes oder einer Gruppe um eine vorgegebene Anzahl Stellen nach links. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits aus der Ausgangsbitfolge nach links herausgeschoben. Die am oberen Ende (MSB) herausgeschobene Anzahl Bits wird am unteren Ende (LSB) hineingeschoben.

	MSB															LSB	
B2 ←	1	1	0	1	1	1	1	1	1	1	0	0	1	0	0	0	←B1

Die SHR-Funktion verschiebt alle Bits eines Wortes oder einer Gruppe um eine vorgegebene Anzahl Stellen nach rechts. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits aus der Ausgangsbitfolge nach rechts herausgeschoben. Die am unteren Ende (LSB) herausgeschobene Anzahl Bits wird am oberen Ende (MSB) hineingeschoben.

	MSB															LSB	
B1 →	1	1	0	1	1	1	1	1	1	1	0	0	1	0	0	0	→B2

Die Stringlänge kann bei beiden Funktionen zwischen 1 und 256 Worten eingestellt werden.

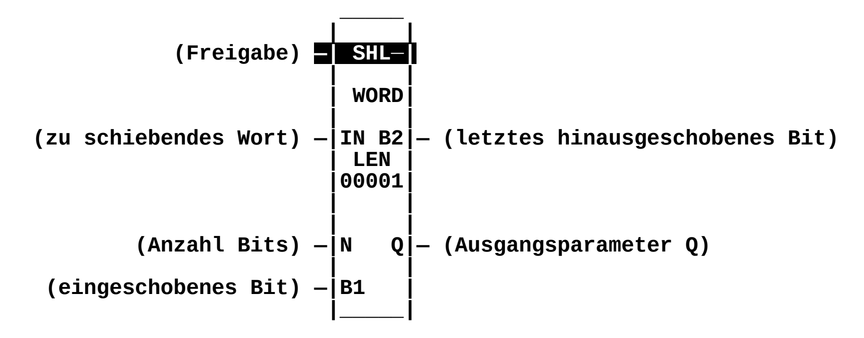
Die Anzahl (N) der Bits, die verschoben werden sollen, muss größer als die Anzahl der in der Bitfolge enthaltenen Bits (LEN * 16) oder Null sein. Ist dies der Fall, wird der Ausgang Q mit Kopien des Eingangsbits (B1) gefüllt und das Eingangsbit wird in den Ausgangs-Stromfluss (B2) kopiert. Ist die Anzahl der zu verschiebenden Bits 0, erfolgt keine Verschiebung. Das Eingangsfeld wird in das Ausgangsfeld kopiert und das Eingangsbit (B1) wird in den Stromfluss kopiert.

Die in den Anfang der Bitfolge eingeschobenen Bits werden über den Eingangsparameter B1 festgelegt. Ist die für die Verschiebung angegebene Anzahl Stellen größer als 1, haben alle eingeschobenen Bits den gleichen Wert (0 oder 1). Es sind hier möglich:

- Der Boolesche Ausgangswert einer anderen Programmfunktion.
- Nur Einsen. Hierzu müssen Sie die Spezialreferenz mit der symbolischen Adresse ALW_ON als Eingabe zu Eingang B1 angeben.
- Nur Nullen. Hierzu müssen Sie die Spezialreferenz mit der symbolischen Adresse ALW_OFF als Eingabe zu Eingang B1 angeben.

Die Funktionen SHL und SHR schalten den Stromfluss nach rechts durch, solange die angegebene Anzahl verschobener Bits nicht Null ist.

Der Funktionsausgang Q ist die verschobene Kopie der Eingangsbitfolge. Wünschen Sie eine Verschiebung der Eingangsbitfolge, muss der Ausgangsparameter Q die gleiche Speicheradresse wie der Eingangsparameter IN verwenden. Die gesamte verschobene Bitfolge wird jedesmal geschrieben, wenn die Funktion Stromfluss empfängt. Ausgangswert B2 ist das letzte hinausgeschobene Bit. Wurden z.B. vier Bits verschoben, dann ist B2 das vierte hinausgeschobene Bit.



Parameter

Parameter	Beschreibung
Freigabe	Die Verschiebung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Wort, das verschoben werden soll.
N	N enthält die Anzahl Plätze (Bits), um die das Feld verschoben werden soll.
B1	B1 enthält den Bitwert, der in das Feld eingeschoben werden soll.
B2	B2 enthält den Wert des letzten aus dem Feld herausgeschobenen Bits.
Q	Der Ausgang Q enthält das erste Wort des verschobenen Feldes.
LEN	LEN ist die Anzahl der Worte in dem verschobenen Feld.

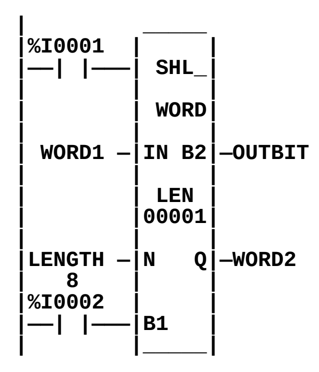
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
B1	•											
B2	•											•
Q		•	•	•	•	•†	•	•	•	•		

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- † Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

Im folgenden Beispiel wird jedesmal, wenn Eingang %I0001 gesetzt wird, die durch die symbolische Adresse WORD2 dargestellte Ausgangsbitfolge als Kopie der um die durch die symbolische Adresse LENGTH angegebene Anzahl Stellen nach links verschobenen Bitfolge WORD1 erstellt. Die resultierenden leeren Bitstellen am Anfang der Ausgangsbitfolge werden mit dem Wert von %I0002 aufgefüllt.



ROL und ROR (WORD)

Mit der ROL-Funktion (rotiere nach links) können alle Bits in einer Bitfolge um eine vorgegebene Anzahl Stellen nach links rotiert werden. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits auf der linken Seite aus der Eingangsbitfolge herausgeschoben und auf der rechten Seite wieder hineingeschoben.

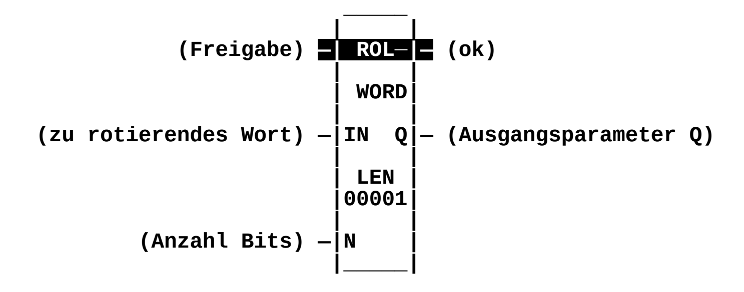
Mit der ROR-Funktion (rotiere nach rechts) können alle Bits in einer Bitfolge um eine vorgegebene Anzahl Stellen nach rechts rotiert werden. Wird die Funktion ausgeführt, wird die angegebene Anzahl Bits auf der rechten Seite aus der Eingangsbitfolge herausgeschoben und auf der linken Seite wieder hineingeschoben.

Die Stringlänge kann bei beiden Funktionen zwischen 1 und 256 Worten eingestellt werden.

Die Anzahl der Plätze, die für die Rotation angegeben wird, muss größer als 0 und kleiner als die Anzahl der in der Bitfolge enthaltenen Bits sein. Ist dies nicht der Fall, erfolgt keine Verschiebung und es findet kein Stromfluss statt.

Die Funktionen ROL und ROR schalten den Stromfluss nach rechts durch, solange die angegebene Anzahl rotierter Bits nicht größer als die Gesamtlänge des Strings oder kleiner als Null ist.

Das Ergebnis wird in die Ausgangsbitfolge Q eingetragen. Wünschen Sie eine Rotation der Eingangsbitfolge, muss der Ausgangsparameter Q die gleiche Speicheradresse wie der Eingangsparameter IN benutzen. Die gesamte rotierte Bitfolge wird bei jedem Zyklus, in dem sie Stromfluss empfängt, geschrieben.



Parameter

Parameter	Beschreibung
Freigabe	Die Rotation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Wort, das rotiert werden soll.
N	N enthält die Anzahl Plätze, um die das Feld rotiert werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Rotation aktiviert ist und die Rotationslänge nicht größer als die Feldlänge ist.
Q	Der Ausgang Q enthält das erste Wort des rotierten Feldes.
LEN	LEN ist die Anzahl der Worte in dem rotierten Feld.

Zulässige Speichertypen

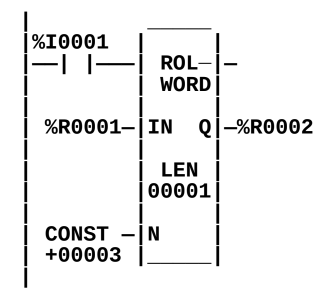
Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
N		•	•	•	•		•	•	•	•	•	
OK	•											•
Q		•	•	•	•	•†	•	•	•	•		

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

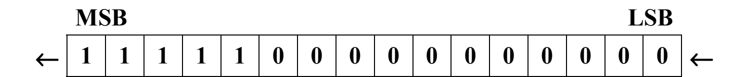
† Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

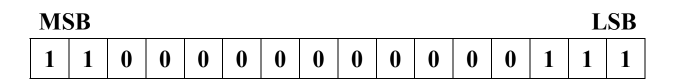
Bei dem folgenden Beispiel wird jedesmal, wenn Eingang %I0001 gesetzt wird, die Eingangsbitfolge %R0001 um 3 Bits rotierend verschoben. Das Ergebnis wird in %R0002 abgelegt. Nachdem die Funktion ausgeführt wurde, bleibt die Eingangsbitfolge %R0001 unverändert. Wird die gleiche Referenz für IN und Q verwendet, findet eine Rotation auf der Stelle statt.



%R0001:



%R0002 (nachdem %I0001 gesetzt wurde):

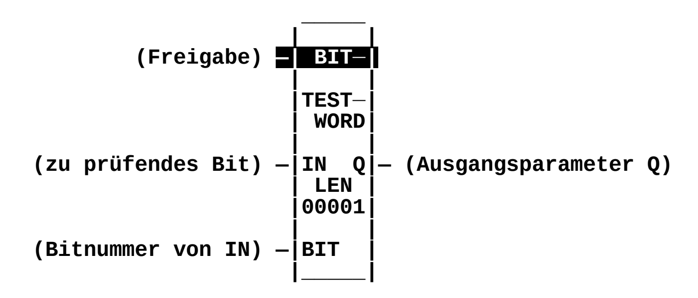


BTST (WORD)

Mit der BTST-Funktion (Bit testen) kann ein Bit in einer Bitfolge daraufhin untersucht werden, ob es momentan 1 oder 0 ist. Das Ergebnis des Tests wird in Ausgang Q eingetragen.

Bei jedem Zyklus, bei dem sie Stromfluss empfängt, setzt die BTST-Funktion ihren Ausgang Q auf den gleichen Zustand, den das angegebene Bit innehat. Wird die Bitnummer nicht durch eine Konstante, sondern durch ein Register angegeben, dann kann der gleiche Funktionsblock in aufeinanderfolgenden Zyklen unterschiedliche Bits testen. Liegt der Wert von BIT außerhalb des zulässigen Bereichs (1 ≤ BIT ≤ (16 * LEN)), dann wird Q auf AUS gesetzt.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden.



Parameter

Parameter	Beschreibung
Freigabe	Der Bittest wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
BIT	BIT enthält die Nummer des Bits von IN, das getestet werden soll. Zulässiger Bereich ist (1 ≤ BIT ≤ (16 * LEN)).
Q	Der Ausgang Q wird durchgeschaltet, wenn das getestete Bit "1" war.
LEN	LEN ist die Anzahl Worte in der getesteten Bitfolge.

Hinweis

Beim Einsatz der Funktionen Bit testen, Bit auf 1 setzen, Bit auf 0 setzen oder Bitposition, werden die Bits von 1 bis 16 nummeriert, ***NICHT*** von 0 bis 15.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
Q	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, das Bit an der in der Referenz PICKBIT angegebenen Adresse getestet. Das Bit ist ein Teil der Bitfolge PRD_CDE. Ist das Bit 1, werden der Ausgang Q und die Spule %Q0001 durchgeschaltet.

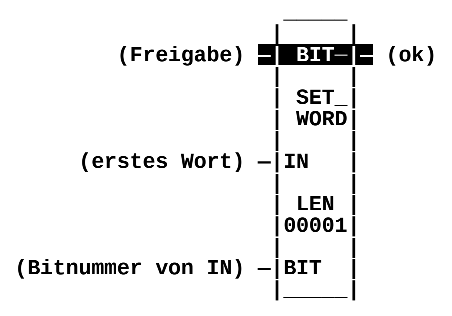


BSET und BCLR (WORD)

Mit der Funktion BSET (Bit auf 1 setzen) kann ein Bit in einer Bitfolge auf 1 gesetzt werden. Mit der Funktion BCLR (Bit auf 0 setzen) kann ein Bit in einer Bitfolge auf 0 gesetzt werden.

Bei jedem Zyklus, bei dem diese Funktionen Stromfluss empfangen, setzen sie das angegebene Bit auf 1 (BSET) bzw. auf 0 (BCLR). Wird die Bitnummer nicht durch eine Konstante, sondern durch eine Variable (Register) angegeben, kann der gleiche Funktionsblock in aufeinanderfolgenden Zyklen unterschiedliche Bits setzen.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden. Die Funktion schaltet den Stromfluss nach rechts durch, wenn der Wert von BIT nicht außerhalb des zulässigen Bereichs (1 ≤ BIT ≤ (16 * LEN)) liegt. Liegt BIT außerhalb dieses Bereichs, dann wird OK auf AUS gesetzt.



Parameter

Parameter	Beschreibung
Freigabe	Die Bitoperation wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
BIT	BIT enthält die Nummer des Bits von IN, das auf 1 oder 0 gesetzt werden soll. Zulässiger Bereich ist (1 ≤ BIT ≤ (16 * LEN)).
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
LEN	LEN ist die Anzahl Worte in der Bitfolge.

Hinweis

Beim Einsatz der Funktionen Bit testen, Bit auf 1 setzen, Bit auf 0 setzen oder Bitposition, werden die Bits von 1 bis 16 nummeriert, ***NICHT*** von 0 bis 15.

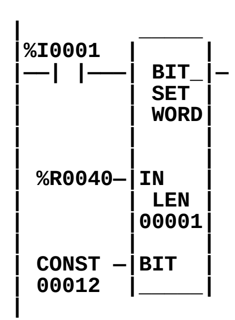
Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	†	•	•	•	•		
BIT		•	•	•	•		•	•	•	•	•	
OK	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
† Nur %SA, %SB, oder %SC; %S kann nicht verwendet werden.

Beispiel

Bei dem folgenden Beispiel wird jedesmal, wenn der Eingang %I0001 gesetzt wird, Bit 12 der bei Referenz %R0040 beginnenden Folge auf 1 gesetzt.



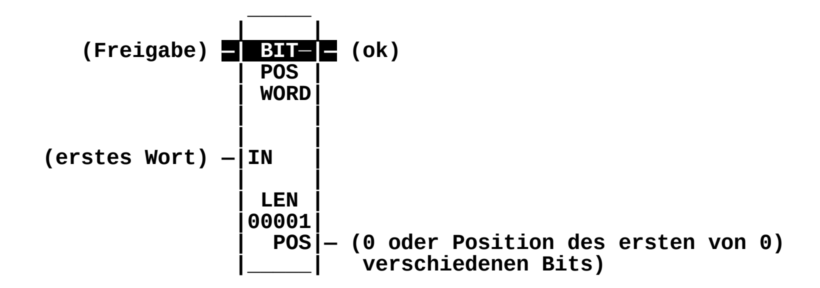
BPOS (WORD)

Mit der BPOS-Funktion kann ein Bit, das auf 1 gesetzt ist, in einer Bitfolge gefunden werden.

Bei jedem Zyklus, bei dem sie Stromfluss empfängt, untersucht die Funktion die bei IN beginnende Bitfolge. Hält die Funktion mit der Suche inne, hat sie entweder ein Bit gefunden, das auf 1 gesetzt wurde, oder sie hat das Ende der Bitfolge erreicht.

POS wird auf die Position innerhalb der Bitfolge eingestellt, in der das erste von Null verschiedene Bit steht. POS wird auf Null gesetzt, wenn kein von Null verschiedenes Bit gefunden wurde.

Es kann eine Bitfolgenlänge zwischen 1 und 256 Worten angegeben werden. Die Funktion schaltet den Stromfluss immer dann nach rechts durch, wenn der Freigabeeingang EIN ist.



Parameter

Parameter	Beschreibung
Freigabe	Die Bitsuche wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält das erste Datenwort, das bearbeitet werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
POS	Wird ein auf 1 gesetztes Bit gefunden, steht hier die Position dieses Bits; wird kein auf 1 gesetztes Bit gefunden, steht hier Null.
LEN	LEN ist die Anzahl Worte in der Bitfolge.

Hinweis

Beim Einsatz der Funktionen Bit testen, Bit auf 1 setzen, Bit auf 0 setzen oder Bitposition, werden die Bits von 1 bis 16 nummeriert, ***NICHT*** von 0 bis 15.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	•	•	•	•	•		
POS		•	•	•	•		•	•	•	•		
OK	•											•

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Wird im nachstehenden Beispiel %I0001 gesetzt, dann wird die bei %M0001 beginnende Bitfolge solange durchsucht, bis ein auf 1 gesetztes Bit gefunden wurde oder 6 Worte durchsucht wurden. Die Spule %Q0001 wird durchgeschaltet. Wird ein auf 1 gesetztes Bit gefunden, dann wird seine Position innerhalb der Bitfolge in %AQ0001 eingetragen. Ist Bit %M0001 0 und Bit %M0002 1, wenn %I0001 gesetzt wird, dann wird in %AQ0001 der Wert 2 eingetragen.

```
|          _____
|%I0001   |     |                                                    %Q0001
|--| |----| BIT_|-------------------------------------------------------( )-
|         |     |
|         | POS_|
|         |     |
|         | WORD|
| %M0001--|IN   |
|         | LEN |
|         |00006|
|         |     |
|         |  POS|--%AQ0001
|         |_____|
|
```

MSKCMP (WORD, DWORD)

Mit der MSKCMP-Funktion (Vergleich mit Maske) können Sie den Inhalt zweier getrennter Bitfolgen miteinander vergleichen und dabei bestimmte Bits maskieren (*diese Funktion ist nur verfügbar für CPUs ab Ausgabestand 4.41*). Die Länge der zu vergleichenden Bitfolgen wird durch den Parameter LEN festgelegt (LEN gibt die Anzahl von 16-Bit-Worten für die Funktion MSKCMPW bzw. die Anzahl von 32-Bit-Worten für die Funktion MSKCMPD an).

Wird über die Freigabelogik Stromfluss an den Freigabeeingang EN angelegt, beginnt die Funktion mit dem Vergleich der jeweils zusammengehörenden Bits in den beiden Bitfolgen. Sie führt diesen Vergleich ihn solange fort, bis entweder ein Unterschied gefunden oder das Ende der Bitfolge erreicht wurde.

Mit dem BIT-Eingang wird die Bitnummer gespeichert, an der der nächste Vergleich beginnen soll eine *0* gibt das erste Bit in der Bitfolge an). Mit dem Ausgang BN wird die Bitnummer gespeichert, bei der der letzte Vergleich stattgefunden hat (eine *1* gibt das erste Bit in der Bitfolge an). Verwenden Sie für BIN und BN die gleiche Referenz, beginnt der Vergleich an der auf einen Unterschied folgenden Bitposition bzw., wenn kein Unterschied gefunden wurde, beim nächsten Aufruf wieder am Anfang.

Soll der nächste Vergleich an einer anderen Stelle der Bitfolge beginnen, können Sie für BIT und BN unterschiedliche Referenzen eingeben. Liegt der Wert von BIT außerhalb der Bitfolge, wird BIT vor dem Beginn des nächsten Vergleichs auf 0 gesetzt.

Alle Bits in I1 und I2 sind gleich

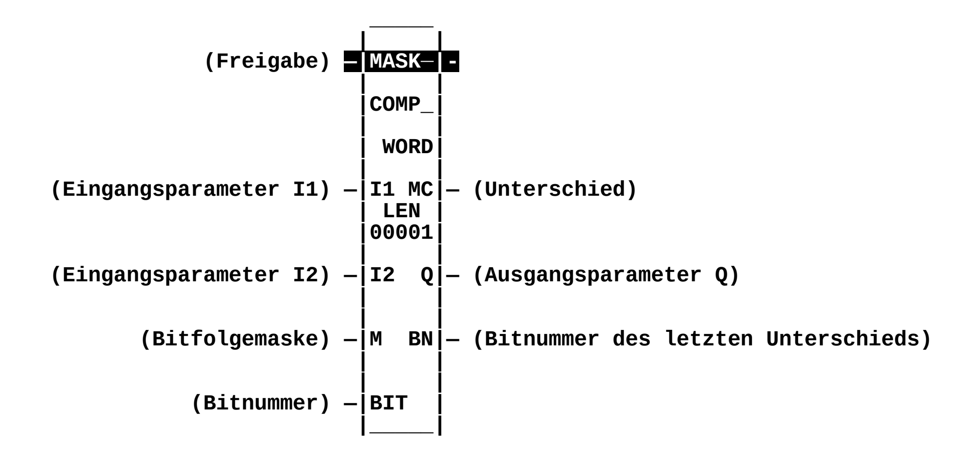
Stimmen alle zusammengehörenden Bits der Bitfolgen I1 und I2 überein, werden der Unterschiedsausgang MC auf 0 und BN auf die höchste Bitnummer der Eingangsfolge gesetzt. Der Vergleich wird dann angehalten. Beim nächsten Aufruf von MSKCMPW wird BN wieder auf 0 rückgesetzt.

Es wurde ein Unterschied festgestellt

Sind zwei miteinander verglichene Bits unterschiedlich, überprüft die Funktion das Bit mit der entsprechenden Nummer in der Bitfolge M (der Maske). Steht an dieser Stelle in der Maske eine *1*, wird der Vergleich fortgesetzt, bis ein weiterer Unterschied gefunden wird oder das Ende der Eingangsbitfolge erreicht ist.

Wird ein Unterschied festgestellt und ist das entsprechende Maskenbit eine 0, führt die Funktion folgende Aktivitäten durch:

1. Das entsprechende Maskenbit in M wird auf 1 gesetzt.
2. Der Unterschiedsausgang MC wird auf 1 gesetzt.
3. Die Ausgangsfolge Q wird so aktualisiert, dass sie dem neuen Inhalt der Maskenfolge M entspricht.
4. Der Bitnummernausgang BN wird auf die Nummer des unterschiedlichen Bits gesetzt.
5. Der Vergleich wird gestoppt.



Parameter

Parameter	Beschreibung
Freigabe	Freigabelogik für Funktionsfreigabe.
I1	Referenz für die erste zu vergleichende Bitfolge.
I2	Referenz für die zweite zu vergleichende Bitfolge.
M	Referenz für die Bitfolgenmaske.
BIT	Referenz für die Bitnummer, bei der der nächste Vergleich beginnen soll.
MC	Anwenderprogramm zur Bestimmung eines Unterschieds.
Q	Ausgangskopie der Maskenbitfolge (M).
BN	Nummer des Bits, bei dem der letzte Vergleich stattfand.
LEN	LEN ist die Anzahl Worte in der Bitfolge.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
I1		o	o	o	o	o	o	•	•	•		
I2		o	o	o	o	o	o	•	•	•		
M		o	o	o	o	o†	o	•	•	•		
BIT		•	•	•	•	•	•	•	•	•	•	
LEN											•‡	
MC	•											•
Q		o	o	o	o	o†	o	•	•	•		
BN		•	•	•	•	•	•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
o Zulässige Referenz nur für WORD-Daten; nicht zulässig für DWORD.
Nur %SA, %SB oder %SC; %S kann nicht verwendet werden.
‡ Maximale Konstante von 4095 für WORD und 2047 für DWORD.

Beispiel

Im folgenden Beispiel wird nach dem ersten Zyklus der Funktionsblock MSKCMPW ausgeführt. %M0001 bis %M0016 werden mit %M0017 bis %M0032 verglichen. Die Bits %M0033 bis %M0048 enthalten die Maskenwerte. Der Wert in %R0001 legt fest, an welcher Bitposition in den beiden Bitfolgen der Vergleich beginnt. Der Inhalt der beiden Referenzen sieht vor der Ausführung des Funktionsblocks wie folgt aus:

(I1) – %M0001 = 6C6Ch =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0

(I2) – %M0017 = 606Fh =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1

(M/Q) – %M0033 = 000Fh =

0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

(BIT/BN) – %R0001 = 0
(MC) – %Q0001 = AUS

Der Inhalt dieser Referenzen sieht nach der Ausführung des Funktionsblocks wie folgt aus:

(I1) – %M0001 =

0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0

(I2) – %M0017 =

0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1

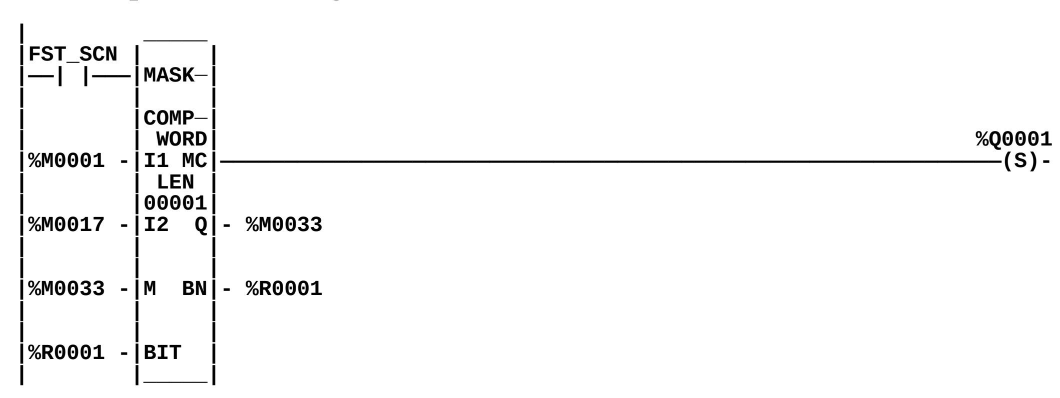
(M/Q) – %M0033 =

0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1

(BIT/BN) – %R0001 = 8

(MC) – %Q0001 = EIN

Kontaktplandarstellung



Beachten Sie, dass wir in dem vorstehenden Beispiel den Kontakt FST_SCN dazu benutzt haben, eine einzige Ausführung zu erzwingen. Im anderen Fall würde der Vergleich mit Maske wiederholt werden und nicht unbedingt das gewünschte Ergebnis erbringen.

Kapitel 9 Datenverschiebefunktionen

Die Datenverschiebefunktionen vermitteln die grundlegenden Möglichkeiten zur Bewegung von Daten. In diesem Kapitel werden die folgenden Datenverschiebefunktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
MOVE	Kopieren	Diese Funktion kopiert Daten bitweise. Die maximale Länge beträgt 256 Worte, bei MOVE_BIT 256 Bits. Daten können ohne vorherige Konvertierung in einen anderen Datentyp kopiert werden.	9-2
BLKMOV	Block kopieren	Diese Funktion kopiert einen Block von sieben Konstanten in eine angegebene Speicheradresse. Die Konstanten werden als Teil der Funktion eingegeben.	9-5
BLKCLR	Block löschen	Diese Funktion überschreibt den Inhalt eines Datenblocks mit Nullen. Sie erlaubt das Löschen eines bit- (%I, %Q, %M, %G oder %T) oder wortstrukturierten (%R, %AI oder %AQ) Speicherbereichs. Maximal sind 256 Worte erlaubt.	9-7
SHFR	Schieberegister	Schiebt ein oder mehrere Datenworte in eine Tabelle. Maximal sind 256 Worte erlaubt.	9-8
BITSEQ	Bitfolgesteuerung	Schiebt eine Bitfolge durch ein Bitfeld. Maximal sind 256 Worte erlaubt.	9-11
COMMREQ	Kommunikationsanforderung	Diese Funktion ermöglicht dem Programm den Datenaustausch mit einem intelligenten Modul, wie zum Beispiel dem Genius-Kommunikationsmodul oder einem programmierbaren Coprozessormodul.	9-14

MOVE (BIT, INT, WORD, REAL)

Mit der MOVE-Funktion können Sie Daten bitweise von einer Stelle zur andern kopieren. Da die Daten im Bitformat kopiert werden, können Quelle und Ziel unterschiedliche Datentypen besitzen.

Die MOVE-Funktion besitzt zwei Ein- und zwei Ausgangsparameter. Erhält die Funktion Stromfluss, kopiert sie Daten bitweise vom Eingangsparameter IN zum Ausgangsparameter Q. Werden Daten von einer Stelle im diskreten Speicher zu einer anderen Stelle kopiert (z.B. vom %I-Speicher zum %T-Speicher), wird die mit den diskreten Speicherelementen verbundene Transitionsinformation aktualisiert um anzuzeigen, ob diskrete Speicherelemente durch die MOVE-Operation ihren Zustand verändert haben. Die Daten an der Eingangsreferenz werden nur verändert, wenn in Quelle und Ziel eine Überlappung stattfindet.

Beim BIT-Typ gelten andere Aspekte. Umfasst ein im Q-Parameter spezifiziertes BIT-Feld nicht alle Bits eines Bytes, werden die mit diesem Byte verknüpften Transitionsbits (die nicht in dem Feld sind) gelöscht, wenn MOVE_BIT Stromfluss erhält.

Der IN-Eingang der Funktion kann entweder eine Referenz für die zu kopierenden Daten oder eine Konstante sein. Wird eine Konstante angegeben, wird der konstante Wert an der von der Ausgangsreferenz angegebenen Stelle eingetragen. Wird zum Beispiel für IN der Wert 4 angegeben, wird diese 4 in die von der Ausgangsreferenz Q angegebene Speicheradresse eingetragen werden. Ist die Länge größer als 1 und wurde eine Konstante angegeben, wird die Konstante ab der durch den Ausgang Q angegebene Speicheradresse über die angegebene Länge eingetragen. Wurden z.B. die Konstante 9 als Eingabewert für IN sowie eine Länge von 4 angegeben, dann wird die 9 in die durch den Ausgang Q festgelegte Speicheradresse sowie die drei darauffolgenden Adressen eingetragen.

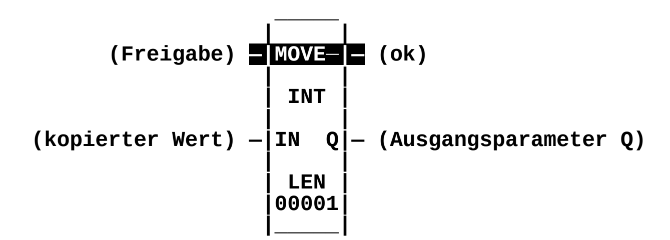
Der Operand LEN gibt an die Anzahl

- Worte die bei MOVE_INT und MOVE_WORD kopiert werden
- Bits, die bei MOVE_BIT kopiert werden
- reelle Werte, die bei MOVE_REAL kopiert werden.

Hinweis

Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Die Funktion schaltet den Stromfluss jedesmal nach rechts durch, wenn sie Stromfluss erhält.



Parameter

Parameter	Beschreibung
Freigabe	Das Kopieren wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält den Wert, der kopiert werden soll. Bei MOVE_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
Q	Der an IN anstehende Wert wird in Q eingetragen wenn die Funktion ausgeführt wird. Bei MOVE_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN gibt an, wieviel Worte oder Bits kopiert werden sollen. Bei MOVE_WORD und MOVE_INT muss LEN zwischen 1 und 256 Worten liegen. Ist IN eine Konstante, muss bei MOVE_BIT LEN zwischen 1 und 16 Bits liegen; in den anderen Fällen zwischen 1 und 256.

Hinweis

Bei den CPUs 351, 352 und 36x unterstützen die Funktionen MOVE_INT und MOVE_WORD keine Überlappung der Parameter IN und Q, wenn die Referenz IN kleiner als die Referenz Q ist. Zum Beispiel ist bei IN=%R0001, Q=%R0004, LEN=5 (Worte) der Inhalt von %R0007 und %R0008 unbestimmt. Mit den Werten Q=%R0001, IN=%R0004, LEN=5 (Worte) ergibt sich jedoch ein gültiger Inhalt.

Beachten Sie auch, dass derzeit nur die CPUs 35x und 36x ab Ausgabestand 9 sowie alle Ausgabestände der CPU352 mit Gleitpunktarithmetik ausgestattet sind. Somit sind diese CPUs die einzigen, die MOVE_REAL ausführen können.

Zulässige Speichertypen

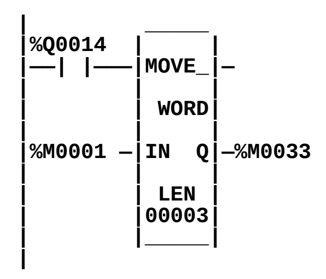
Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	o	•	•	•	•	•	
OK	•											•
Q		•	•	•	•	o†	•	•	•	•		

<u>Hinweis</u>: Bei REAL-Daten sind nur die Typen %R, %AI und %AQ zulässig.

- • Zulässige Referenz für BIT-, INT-, oder WORD-Daten oder Platz, an dem Strom durch die Funktion fließen kann. Bei MOVE_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.
- o Zulässige Referenz nur für BIT- oder WORD-Daten; nicht zulässig für INT.
- † Nur %SA, %SB, %SC; %S kann nicht verwendet werden.

Beispiel 1

Wird der Freigabeeingang %Q00014 EIN, werden 48 Bits von der Speicheradresse %M0001 in die Speicheradresse %M0033 kopiert. Die Verschiebung wird korrekt durchgeführt, obwohl bei Quelle und Ziel 16 Bits überlappen (außer bei den CPUs 351 und 352, wie zuvor erwähnt).



Vor Ausführung der MOVE-Funktion:

INPUT (%M0001 bis %M0048)

1

%M0016	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0032	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0048	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Nach Ausführung der MOVE-Funktion:

INPUT (%M0033 bis %M0080)

33

%M0048	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
%M0064	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
%M0080	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Beispiel 2

Wird in diesem Beispiel %I0001 gesetzt, werden die drei Bits %M0001, %M0002 und %M0003 nach %M0100, %M0101 und %M0102 verschoben. Die Spule %Q0001 wird durchgeschaltet.

```
|         _____
|%I0003  |     |                                                   %Q0001
|--| |---|MOVE_|---------------------------------------------------( )-
|        |     |
|        | BIT |
|        |     |
| %M0001-|IN  Q|-%M0100
|        |     |
|        | LEN |
|        |00003|
|        |_____|
|
```

BLKMOV (INT, WORD, REAL)

Mit der BLKMOV-Funktion können Sie einen Block von sieben Konstanten zu einer angegebenen Speicheradresse kopieren.

Hinweis

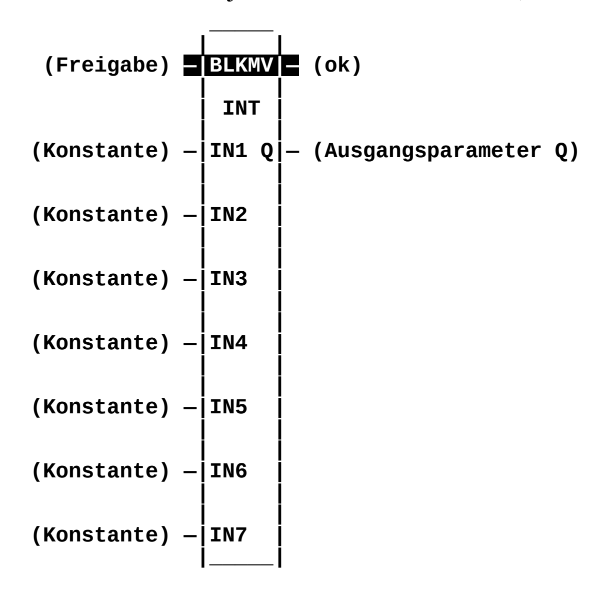
Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Die BLKMOV-Funktion besitzt acht Ein- und zwei Ausgangsparameter. Erhält die Funktion Stromfluss, kopiert sie die Konstanten in einen zusammenhängenden Speicherbereich, der bei der im Ausgang Q spezifizierten Adresse beginnt. Ausgang Q kann nicht der Eingang einer anderen Programmfunktion sein.

Hinweis

Bei BLKMOV_INT werden die Werte von IN1 — IN7 als Dezimalwerte mit Vorzeichen angezeigt. Bei BLKMOV_WORD werden IN1 — in Hexadezimal-darstellung angezeigt. Bei BLKMOV_REAL werden IN1— IN7 als reelle Zahlen angezeigt.

Die Funktion schaltet den Stromfluss jedesmal nach rechts durch, wenn sie Stromfluss erhält.



Parameter

Parameter	Beschreibung
Freigabe	Der Block wird kopiert, wenn die Funktion freigegeben ist.
IN1— IN7	IN1-IN7 enthalten sieben Konstanten.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
Q	Ausgang Q enthält die erste ganze Zahl des kopierten Feldes. IN1 wird zu Q kopiert.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN1 — IN7											•	
OK	•											•
Q		•	•	•	•	o†	•	•	•	•		

Hinweis: Bei REAL-Daten sind nur die Typen %R, %AI und %AQ zulässig.
• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
o Zulässige Referenz nur für WORD-Daten; nicht zulässig für INT oder REAL.
† Nur %SA, %SB, %SC; %S kann nicht verwendet werden.

Hinweis

Derzeit sind nur die CPUs 35x und 36x ab Ausgabestand 9 sowie alle Ausgabestände der CPU352 mit Gleitpunktarithmetik ausgestattet. Somit sind diese CPUs die einzigen, die BLKMV_REAL ausführen können.

Beispiel

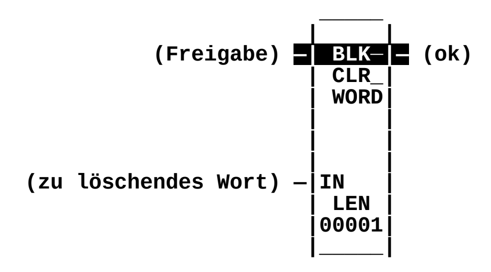
Beim folgenden Beispiel kopiert die BLKMOV-Funktion die sieben Eingangskonstanten in die Speicheradressen %R0010 bis %R0016, wenn der durch die symbolische Adresse FST_SCN gekennzeichnete Freigabeeingang EIN ist.

```
|          _____
|FST_SCN  |     |
|--| |----|BLKMV|-
|         | INT |
|         |     |
| CONST  -|IN1 Q|- %R0010
| +32767  |     |
|         |     |
| CONST  -|IN2  |
| -32768  |     |
|         |     |
| CONST  -|IN3  |
| +00001  |     |
|         |     |
| CONST  -|IN4  |
| +00002  |     |
|         |     |
| CONST  -|IN5  |
| -00002  |     |
|         |     |
| CONST  -|IN6  |
| -00001  |     |
|         |     |
| CONST  -|IN7  |
| +00001  |     |
|         |_____|
|
```

BLKCLR (WORD)

Mit der BLKCLR-Funktion (Block löschen) kann ein spezifizierter Datenblock mit Nullen gefüllt werden.

Die BLKCLR-Funktion besitzt zwei Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluss, schreibt sie die Speicheradresse, die mit der durch IN festgelegten Adresse beginnt, mit Nullen voll. Stehen die gelöschten Daten im diskreten Speicher (%I, %Q, %M, %G oder %T), wird die mit den Referenzen verbundene Transitionsinformation ebenfalls gelöscht. Die Funktion schaltet den Stromfluss nach rechts durch.



Parameter

Parameter	Beschreibung
Freigabe	Das Feld wird gelöscht, wenn die Funktion freigegeben ist.
IN	IN enthält den ersten Wert des Feldes, das gelöscht werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
LEN	LEN muss zwischen 1 und 256 Worten sein.

Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•	•†	•	•	•	•		
OK	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann. † Nur %SA, %SB, %SC; %S kann nicht verwendet werden.

Beispiel

Im folgenden Beispiel werden beim Einschalten 32 Worte des %Q-Speichers (512 Punkte), beginnend mit %Q0001, mit Nullen gefüllt.

```
|          _____
|FST_SCN  |     |
|--| |----| BLK_|-
|         |     |
|         | CLR_|
|         | WORD|
|         |     |
| %Q0001--|IN   |
|         | LEN |
|         |00032|
|         |_____|
|
```

SHFR (BIT, WORD)

Mit der Schieberegisterfunktion (SHFR) können ein oder mehrere Datenworte oder Datenbits von einer Referenzadresse in einen angegebenen Speicherbereich verschoben werden. So kann zum Beispiel ein Wort in einen Speicherbereich mit einer angegebenen Länge von fünf Worten eingeschoben werden. Als Ergebnis dieses Vorgangs wird am Ende des Speicherbereichs ein anderes Datenwort herausgeschoben.

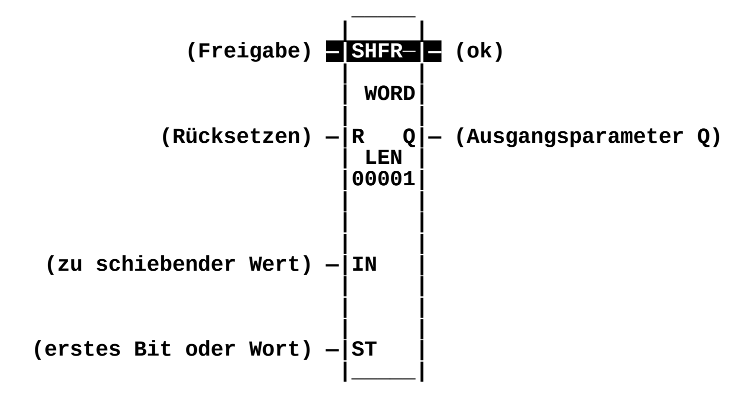
Hinweis

Bei der Zuweisung von Referenzadressen können überlappende Ein- und Ausgangsreferenzadressbereiche in Mehrwortfunktionen zu unvorhersehbaren Reaktionen führen.

Die SHFR-Funktion besitzt vier Ein- und zwei Ausgangsparameter. Der Rücksetzeingang (R) hat Priorität über den Funktionsfreigabeeingang. Ist der Rücksetzeingang aktiv, werden ab dem Schieberegister (ST) über die für LEN angegebene Länge alle Referenzen mit Nullen gefüllt.

Erhält die Funktion Stromfluss, ohne dass der Rücksetzeingang aktiv ist, werden die Daten im Schieberegister um ein Element (entweder ein Bit oder ein Wort) zur nächst höheren Referenz hin verschoben. Das letzte Element des Schieberegisters wird in Q geschoben. Die höchste Referenz des Schieberegisterelements von IN wird in das nun freie Element ab ST eingeschoben. Auf den Inhalt des Schieberegisters kann im gesamten Programmablauf zugegriffen werden, da er an absoluten Adressen im logisch adressierbaren Speicherbereich liegt.

Die SHFREG-Funktion schaltet den Stromfluss nach rechts durch, wenn sie über die Freigabelogik Stromfluss erhält.



Parameter

Parameter	Beschreibung
Freigabe	Die Verschiebung wird durchgeführt, wenn der Freigabeeingang aktiviert ist und R nicht.
R	Das Schieberegister bei ST wird mit Nullen gefüllt, wenn R aktiviert wird.
IN	IN enthält den Wert, der ins erste Bit oder Wort des Schieberegisters eingeschoben werden soll. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
ST	ST enthält das erste Wort oder Bit des Schieberegisters. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben und R nicht freigegeben ist.
Q	Der Ausgang Q enthält das aus dem Schieberegister herausgeschobene Bit oder Wort. Bei SHFR_BIT kann eine beliebige diskrete Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN legt die Länge des Schieberegisters fest. Für SHFR_WORD muss LEN zwischen 1 und 256 Worten liegen. Für SHFR_BIT muss LEN zwischen 1 und 256 Bits liegen.

Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
R	•											
IN		•	•	•	•	•	•	•	•	•	•	
ST		•	•	•	•	•†	•	•	•	•		
OK	•											•
Q		•	•	•	•	•†	•	•	•	•		

• Zulässige Referenz für BIT- oder WORD-Daten oder Platz, an dem Strom durch die Funktion fließen kann. Bei SHFR_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.

† Nur %SA, %SB, %SC; %S kann nicht verwendet werden.

Beispiel 1

Bei dem folgenden Beispiel wird die Funktion auf die Registerspeicher-Adressen %R0001 bis %R0100 angewandt. Die Schieberegisterworte werden auf Null gesetzt, wenn die Rücksetzreferenz CLEAR aktiv ist.

Ist die Referenz NXT_CYC aktiv und ist CLEAR nicht aktiv, dann wird das Wort von Adresse %Q0033 der Ausgangs-Zustandstabelle in das Schieberegister geschoben. Das aus dem Schieberegister von %R0100 herausgeschobene Wort wird im Ausgang %M0005 abgelegt.

```
|          _____
|NXT_CYC  |     |
|--| |----|SHFR_|-
|         |     |
|         | WORD|
| CLEAR   |     |
|--| |----|R   Q|-%M0005
|         | LEN |
|         |00100|
|         |     |
|         |     |
|         |     |
| %Q0033--|IN   |
|         |     |
|         |     |
| %R0001--|ST   |
|         |_____|
|
```

Beispiel 2

Bei diesem Beispiel wird die Funktion auf die Speicheradressen %M0001 bis %M0100 angewandt. Ist die Rücksetz-Referenz CLEAR aktiv, dann füllt die SHFR-Funktion %M0001 bis %M0100 mit Nullen.

Ist NXT_CYC aktiv und CLEAR inaktiv, dann verschiebt die SHFR-Funktion die Daten aus %M0001 bis %M0100 um ein Bit nach unten. Das Bit in %Q0033 wird nach %M0001 geschoben, während das aus %M0100 herausgeschobene Bit in %M0200 eingetragen wird.

```
|          _____
|NXT_CYC  |     |
|--| |----|SHFR_|-
|         |     |
|         | BIT |
| CLEAR   |     |
|--| |----|R   Q|-%M0200
|         | LEN |
|         |00100|
|         |     |
|         |     |
|         |     |
| %Q0033--|IN   |
|         |     |
|         |     |
| %M0001--|ST   |
|         |_____|
|
```

BITSEQ (BIT)

Mit der Bitfolgesteuerung (BITSEQ) kann eine Bitfolge durch eine Bitfeld verschoben werden. Die BITSEQ-Funktion besitzt fünf Ein- und einen Ausgangsparameter. Das Ergebnis der Funktion hängt von dem vorhergehenden Wert des Parameters EN ab:

R aktueller Durchlauf	EN voriger Durchlauf	EN aktueller Durchlauf	BITSEQ-Funktion
AUS	AUS	AUS	BITSEQ-Funktion wird nicht ausgeführt.
AUS	AUS	EIN	BITSEQ-Funktion erhöht/erniedrigt um 1.
AUS	EIN	AUS	BITSEQ-Funktion wird nicht ausgeführt.
AUS	EIN	EIN	BITSEQ-Funktion wird nicht ausgeführt.
EIN	EIN/AUS	EIN/AUS	BITSEQ-Funktion wird rückgesetzt.

Der Rücksetzeingang (R) überschreibt den Freigabeeingang (EN) und bewirkt immer ein Rücksetzen der BITSEQ-Funktion. Ist der Rücksetzeingang aktiv, dann wird die aktuelle Schrittnummer auf den über den Schrittnummerparameter eingegebenen Wert gesetzt. Wurde keine Schrittnummer eingegeben, wird der Schritt auf 1 gesetzt. Mit Ausnahme des Bits, auf das der aktuelle Schritt zeigt und das auf 1 gesetzt wird, werden sämtliche Bits in der Bitfolgesteuerung auf 0 gesetzt.

Ist EN aktiv und R nicht, dann wird das Bit gelöscht, auf das der aktuelle Schritt zeigt. Die aktuelle Schrittnummer wird je nach Richtungsparameter entweder erhöht oder erniedrigt. Danach wird das Bit, auf das die neue Schrittnummer zeigt, auf 1 gesetzt.

- Die Schrittnummer wird auf 1 rückgesetzt, wenn sie beim Erhöhen außerhalb des Bereichs (1 ≤ Schrittnummer ≤ LEN) gerät.
- Die Schrittnummer wird auf LEN rückgesetzt, wenn sie beim Erniedrigen außerhalb des Bereichs (1 ≤ Schrittnummer ≤ LEN) gerät.

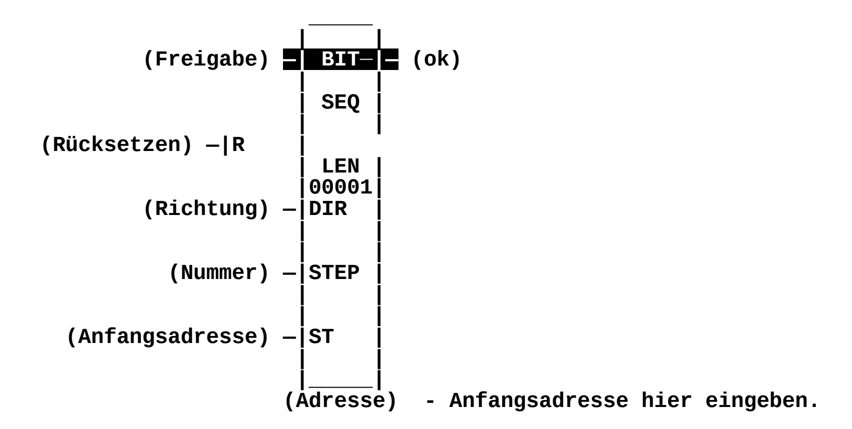
Die Angabe des ST-Parameters ist nicht zwingend. Wird er nicht verwendet, dann arbeitet die BITSEQ-Funktion wie beschrieben, es werden jedoch keine Bits gelöscht oder gesetzt. Die BITSEQ-Funktion schaltet lediglich die aktuelle Schrittnummer durch den zulässigen Bereich hindurch.

Speicherbedarf der Bitfolgesteuerung

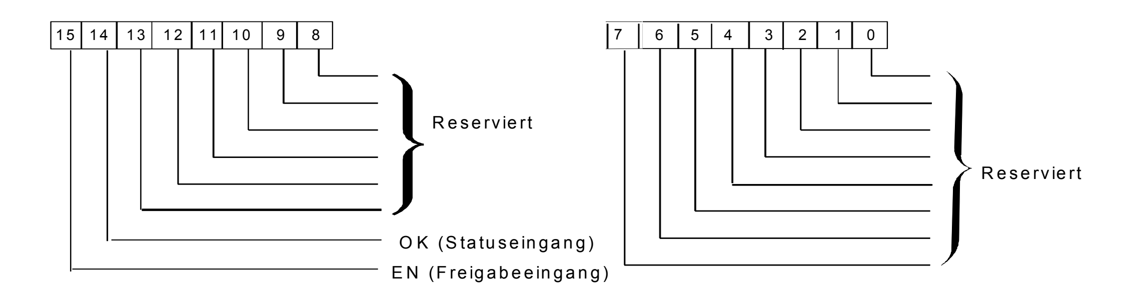
Jede Bitfolgesteuerung belegt im %R-Speicher drei Worte (Register), um die folgenden Daten abzuspeichern:

aktuelle Schrittnummer	Wort 1
Länge der Bitfolge (in Bits)	Wort 2
Steuerwort	Wort 3

Bei der Eingabe der BITSEQ-Funktion müssen Sie eine Anfangsadresse für diese drei Worte (Register) unmittelbar unterhalb der Funktionsgraphik eingeben (siehe Beispiel auf der nächsten Seite).



Im Steuerwort werden die Zustände der Booleschen Ein- und Ausgänge des zugehörigen Funktionsblocks in folgendem Format abgelegt:



Hinweis

Bits 0 bis 13 werden nicht verwendet. Im Parameter STEP müssen die Bits als 1 bis 16 eingegeben werden, ***NICHT*** als 0 bis 15.

Parameter

Parameter	Beschreibung
Adresse	Gibt die Speicheradresse von aktuellem Schritt und Länge sowie der letzten Freigabe- und OK-Zustände an.
Freigabe	Die Bitfolge wird verschoben, wenn die Funktion freigegeben wird, nachdem sie beim letzten Zyklus nicht freigegeben war und R nicht durchgeschaltet ist.
R	Wird R aktiviert, dann wird die Schrittnummer der BITSEQ-Funktion auf den Wert in STEP (Standard = 1) gesetzt und mit Ausnahme des aktuellen Schrittnummernbits werden alle Bits der Folge auf Null gesetzt.
DIR	Wird DIR aktiviert, dann wird die Schrittnummer vor dem Verschieben um 1 erhöht. Ist DIR nicht durchgeschaltet, dann wird sie erniedrigt.
STEP	Die Schrittnummer wird auf diesen Wert eingestellt, wenn R aktiviert wird.
ST	ST enthält das erste Wort der Bitfolgesteuerung.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion freigegeben ist.
LEN	LEN muss zwischen 1 und 256 Bits sein.

Hinweis

Bei der BITSEQ-Funktion prüft die Spulenverwendungs-Überprüfungsfunktion 16 Bits des ST-Parameters, selbst wenn LEN kleiner als 16 ist.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Adresse								•				
Freigabe	•											
R	•											
DIR	•											
STEP		•	•	•	•		•	•	•	•	•	•
ST		•	•	•	•	•†	•	•	•	•		•
OK	•											•

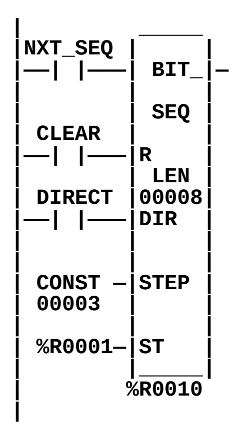
- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

† Nur %SA, %SB, %SC; %S kann nicht verwendet werden.

Beispiel

Beim folgenden Beispiel verwendet die BITSEQ-Funktion den Registerspeicher %R0001. Die statischen Daten sind in den Registern %R0010, %R0011 und %R0012 abgelegt. Ist CLEAR aktiv, dann wird die Funktion rückgesetzt und der aktuelle Schritt wird auf Schrittnummer 3 eingestellt. Die ersten acht Bits von %R0001 werden auf Null gesetzt.

Ist NXT_SEQ aktiv und CLEAR nicht, dann wird das Bit für Schrittnummer 3 gelöscht und das Bit für Schrittnummer 2 oder 4 (hängt davon ab, ob DIR durchgeschaltet ist) wird gesetzt.



COMMREQ

Die COMMREQ-Funktion wird eingesetzt, wenn das Programm mit einem intelligenten Modul (z.B. Genius-Kommunikationsmodul oder programmierbares Coprozessormodul) Daten austauschen will.

Hinweis

Auf den folgenden Seiten wird lediglich das Format der COMMREQ-Funktion dargestellt. Für jeden Modultyp brauchen Sie zusätzliche Angaben bei der Programmierung der COMMREQ-Funktion. Diese Programmieranforderungen finden Sie in der Dokumentation zu den einzelnen Modulen.

Die COMMREQ-Funktion besitzt drei Ein- und einen Ausgangsparameter. Erhält eine COMMREQ-Funktion Stromfluss, dann wird ein Befehlsdatenblock an das intelligente Modul geschickt. Dieser Befehlsblock beginnt mit der über den Parameter IN spezifizierten Referenz. Das Gerät, mit dem Daten ausgetauscht werden sollen, wird durch seine in SYSID angegebene Chassis- und Steckplatznummer identifiziert.

Die COMMREQ-Funktion sendet entweder eine Meldung und wartet auf Antwort oder sie sendet eine Meldung und fährt ohne Antwort fort. Gibt der Befehlsblock an, dass das Programm nicht auf eine Antwort wartet, dann wird der Inhalt des Befehlsblocks zu dem Empfangsgerät gesendet und die Programmausführung direkt fortgesetzt (der Zeitüberwachungswert wird ignoriert). Diese Betriebsart ist der **NOWAIT**-Modus.

Gibt der Befehlsblock an, dass das Programm auf eine Antwort wartet, wird der Inhalt des Befehlsblocks zu dem Empfangsgerät gesendet und die CPU wartet auf eine Antwort. Die maximale Wartezeit wird im Befehlsblock angegeben. Antwortet das Gerät nicht innerhalb dieser Zeit, wird die Programmausführung fortgesetzt. Diese Betriebsart ist der **WAIT**-Modus.

Der Funktionsfehlerausgang (FT) wird gesetzt, wenn:

1. die angegebene Zieladresse nicht existiert (SYSID);
2. die angegebene Task für das Gerät nicht zulässig ist (TASK);
3. die Datenlänge 0 beträgt;
4. die Status-Pointeradresse (Teil des Befehlsblocks) des Geräts nicht existiert. Ursache hierfür kann eine fehlerhafte Wahl des Speichertyps sein oder eine Adresse innerhalb dieses Speichertyps, die außerhalb des zulässigen Bereichs liegt.

Befehlsblock

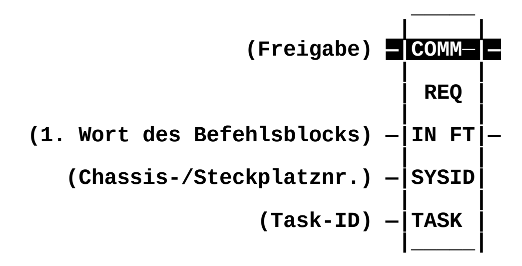
Der Befehlsblock liefert dem intelligenten Modul Informationen zu dem auszuführenden Befehl.

Die Adresse des Befehlsblocks wird für den IN-Eingang zur COMMREQ-Funktion angegeben. Bei dieser Adresse kann es sich um einen beliebigen wortstrukturierten Speicherbereich (%R, %AI oder %AQ) handeln. Die Länge des Befehlsblocks ist von der zu dem Gerät geschickten Datenmenge abhängig.

Der Befehlsblock besitzt folgende Struktur:

Länge (in Worten)	Adresse
Merker "WAIT/NO WAIT"	Adresse + 1
Statuspointerspeicher	Adresse + 2
Statuspointeroffset	Adresse + 3
Pausenüberwachungswert	Adresse + 4
Maximale Kommunikationszeit	Adresse + 5
Datenblock	Adresse + 6 bis Adresse + 133

Die für den Befehlsblock erforderlichen Angaben können mit der entsprechenden Programmierfunktion in die angegebenen Speicherbereiche eingetragen werden.



Parameter

Parameter	Beschreibung
Freigabe	Die Kommunikationsanforderung wird ausgeführt, wenn die Funktion aktiviert ist.
IN	IN enthält das erste Wort des Befehlsblocks.
SYSID	SYSID enthält Chassisnummer (höchstwertiges Byte) und Steckplatznummer (niedrigstwertiges Byte) des Zielgerätes.
TASK	Dieser Parameter enthält die Task-Kennung vom Prozess des Zielgeräts.
FT	FT wird durchgeschaltet, wenn bei der Bearbeitung der COMMREQ ein Fehler erkannt wird.

Zulässige Speichertypen

Parameter	Stromfluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN								•	•	•		
SYSID		•	•	•	•		•	•	•	•	•	
TASK								•	•	•	•	
FT	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Ist im folgenden Beispiel der Freigabeeingang %M0020 EIN, wird der ab Adresse %R0016 abgelegte Befehlsblock zur Kommunikations-Task 1 des Geräts übertragen, das in Steckplatz 2 von Chassis 1 in der SPS eingebaut ist. Tritt ein Fehler bei der Bearbeitung der COMMREQ-Funktion auf, wird %Q0100 durchgeschaltet.



Hinweis

Bei Systemen ohne Erweiterungschassis muss SYSID des Hauptchassis Null sein.

Kapitel 10 Tabellenfunktionen

Mit den Tabellenfunktionen können Sie folgende Funktionen ausführen:

Abkürzung	Funktion	Beschreibung	Seite
ARRAY_MOVE	Feld kopieren	Kopiert eine vorgegebene Anzahl Datenelemente von einem Quellfeld zu einem Zielfeld.	10-2
SRCH_EQ	Suche gleiche	Sucht nach allen Feldwerten, die zu einem angegebenen Wert gleich sind.	10-6
SRCH_NE	Suche ungleiche	Sucht nach allen Feldwerten, die von einem angegebenen Wert verschieden sind.	10-6
SRCH_GT	Suche größer als	Sucht nach allen Feldwerten, die größer als ein angegebener Wert sind.	10-6
SRCH_GE	Suche größer als/gleich	Sucht nach allen Feldwerten, die größer als ein angegebener Wert oder gleich sind.	10-6
SRCH_LT	Suche kleiner als	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert sind.	10-6
SRCH_LE	Suche kleiner als/gleich	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert oder gleich sind.	10-6

Die maximal zulässige Länge bei diesen Funktionen beträgt 32.767 Byte oder Worte bzw. 262.136 Bits (Bits nur für ARRAY_MOVE).

Tabellenfunktionen können auf folgende Datentypen angewandt werden:

Datentyp	Beschreibung
INT	Ganze Zahl mit Vorzeichen.
DINT	Doppeltgenaue ganze Zahl mit Vorzeichen.
BIT *	Bit.
BYTE	Byte.
WORD	Wort.

* Nur für ARRAY_MOVE.

Die Standardbelegung (ganze Zahl mit Vorzeichen) kann nach Anwahl der Funktion verändert werden. Um andere Datentypen oder Daten mit zwei unterschiedlichen Typen vergleichen zu können, müssen Sie zunächst die entsprechende Datenkonvertierungsfunktion (beschrieben in Kapitel 11, "Umwandlungsfunktionen") einsetzen, um eine Umwandlung in die oben genannten Datentypen vorzunehmen.

ARRAY_MOVE (INT, DINT, BIT, BYTE, WORD)

Mit der Funktion ARRAY_MOVE (Feld kopieren) können Sie eine angegebene Anzahl Datenelemente von einem Quellfeld in ein Zielfeld kopieren.

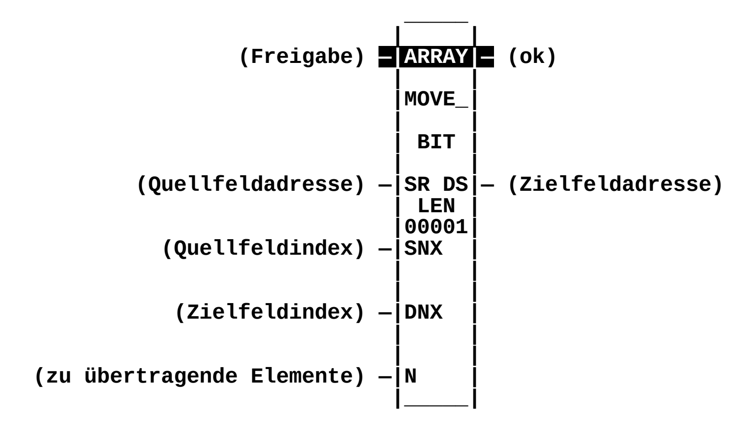
Die Funktion ARRAY_MOVE besitzt fünf Ein- und zwei Ausgangsparameter. Empfängt die Funktion Stromfluss, wird ab der indizierten Adresse (SR + SNX — 1) die Anzahl der Datenelemente in der Zählanzeige (N) aus dem Eingangsfeld herausgezogen. Die Datenelemente werden in das Ausgangsfeld mit der indizierten Adresse (DS + DNX — 1) geschrieben. Der Operand LEN gibt an, aus wieviel Elementen die Felder bestehen.

Wird bei ARRAY_MOVE_BIT für die Parameter der Anfangsadresse von Quell- und/oder Zielfeld wortstrukturierter Speicher gewählt, ist das niedrigstwertige Bit des angegebenen Wortes das erste Bit des Feldes. Der angezeigte Wert enthält unabhängig von der Feldlänge immer 16 Bits.

Die Indizes einer ARRAY_MOVE-Anweisung sind auf Basis 1. Beim Einsatz von ARRAY_MOVE darf kein Element außerhalb der Quell- und Zielfelder (die durch ihre Anfangsadresse und Länge spezifiziert wurden) adressiert werden.

Der OK-Ausgang empfängt Stromfluss, wenn nicht einer der folgenden Zustände vorliegt:

- Der Freigabeeingang ist AUS.
- (N + SNX – 1) ist größer als LEN.
- (N + DNX – 1) ist größer als LEN.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
SR	SR enthält die Anfangsadresse des Quellfeldes. Für ARRAY_MOVE_ BIT kann eine beliebige Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
SNX	SNX enthält den Index des Quellfeldes.
DNX	DNX enthält den Index des Zielfeldes.
N	N bietet eine Zählanzeige.
OK	Der OK-Ausgang wird durchgeschaltet, wenn der Freigabeeingang aktiviert ist.
DS	DS enthält die Anfangsadresse des Zielfeldes. Für ARRAY_MOVE_ BIT kann eine beliebige Referenz verwendet werden, die nicht im Byteraster zu liegen braucht. Es werden jedoch im Online-Betrieb 16 Bits ab der angegebenen Referenzadresse angezeigt.
LEN	LEN gibt die Anzahl Elemente ab SR und DS an, aus denen die Felder bestehen.

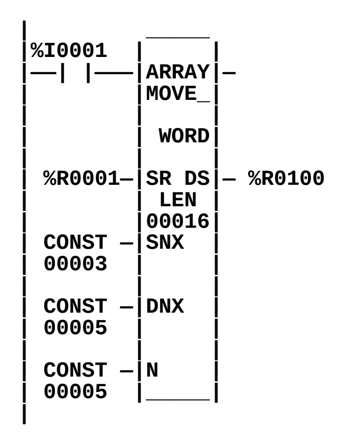
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
SR		o	o	o	o	Δ†	o	•	•	•		
SNX		•	•	•	•		•	•	•	•	•	
DNX		•	•	•	•		•	•	•	•	•	
N		•	•	•	•		•	•	•	•	•	
OK	•											•
DS		o	o	o	o	†	o	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
Bei ARRAY_MOVE_BIT brauchen die diskreten Anwenderreferenzen %I, %Q, %M und %T nicht im Byteraster zu liegen.
o Zulässige Referenz nur für INT, BIT, BYTE oder WORD, nicht zulässig für DINT.
Δ Zulässiger Datentyp nur für BIT, BYTE oder WORD, nicht zulässig für INT oder DINT.
† Nur %SA, %SB, %SC;__ %S kann nicht verwendet werden.

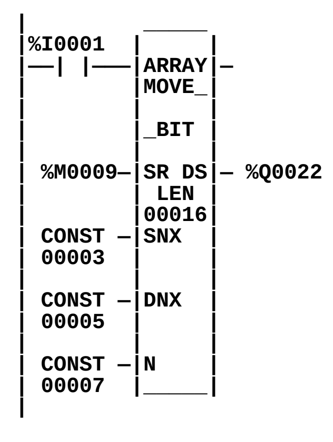
Beispiel 1

In diesem Beispiel werden %R0003 — %R0007 des Feldes %R0001 — %R0016 gelesen und in %R0104 — %R0108 des Feldes %R0100 — %R0115 geschrieben.



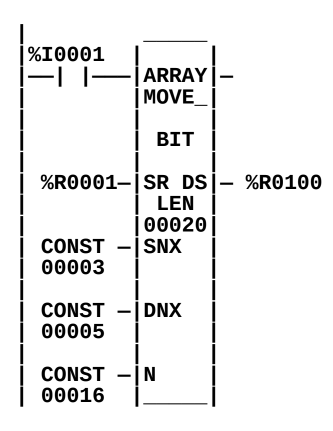
Beispiel 2

Unter Benutzung von Bitspeicher für SR und DS wird %M0011 — %M0017 des Feldes %M0009 — %M0024 gelesen und in %Q0026 — %Q0032 des Feldes %Q0022 — %Q0037 geschrieben.



Beispiel 3

Unter Benutzung von Wortspeicher für SR und DS wird aus dem Feld mit allen sechzehn Bits von %R0001 und vier Bits von %R0002 der Teil vom drittniedrigsten Bit von %R0001 bis einschließlich dem zweitniedrigsten Bit von %R0002 gelesen und dann in den Bereich vom fünftniedrigsten Bit von %R0100 bis einschließlich dem viertniedrigsten Bit von %R0101 des Feldes geschrieben, das alle 16 Bits von %R0100 und vier Bits von %R0101 enthält.



Suchfunktionen

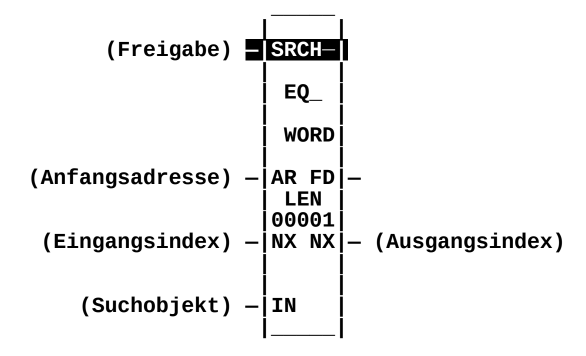
Mit den nachstehend aufgeführten Suchfunktionen können Sie nach Feldwerten suchen, gemäß der verwendeten Operationen.

Abkürzung	Funktion	Beschreibung
SRCH_EQ	Suche gleiche	Sucht nach allen Feldwerten, die zu einem angegebenen Wert gleich sind.
SRCH_NE	Suche ungleiche	Sucht nach allen Feldwerten, die von einem angegebenen Wert verschieden sind.
SRCH_GT	Suche größer als	Sucht nach allen Feldwerten, die größer als ein angegebener Wert sind.
SRCH_GE	Suche größer als/gleich	Sucht nach allen Feldwerten, die größer als ein angegebener Wert oder gleich sind.
SRCH_LT	Suche kleiner als	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert sind.
SRCH_LE	Suche kleiner als/gleich	Sucht nach allen Feldwerten, die kleiner als ein angegebener Wert oder gleich sind.

Jede Funktion besitzt vier Ein- und zwei Ausgangsparameter. Empfängt die Funktion Stromfluss, wird das Feld ab (AR + Eingang NX) durchsucht. Der Startpunkt setzt sich zusammen aus der Anfangsadresse des Feldes (AR) plus dem Index zu diesem Feld (Eingang NX).

Die Suche wird durchgeführt, bis das Feldelement des Suchobjekts (IN) gefunden oder das Feldende erreicht ist. Wird ein Feldelement gefunden, wird der Ausgangsparameter (FD) durchgeschaltet und der Ausgangsparameter (Ausgang NX) auf die Relativposition dieses Elements im Feld gesetzt. Wird vor Erreichen des Feldendes kein passendes Feldelement gefunden, wird Ausgangsparameter FD auf AUS gesetzt und der Ausgangsparameter (Ausgang NX) auf Null.

Zulässige Werte für Eingang NX sind 0 bis LEN — 1. Um die Suche beim ersten Element zu beginnen, muss NX auf Null gesetzt werden. Da bei der Ausführung dieser Wert um 1 erhöht wird, ergibt sich für Ausgang NX ein Wertebereich zwischen 1 und LEN. Liegt der Wert von Eingang NX außerhalb des Bereichs (< 0 oder ≥ LEN), wird sein Wert auf den Standardwert 0 gesetzt.



Parameter

Parameter	Beschreibung
Freigabe	Die Operation wird durchgeführt, wenn die Funktion freigegeben ist.
AR	AR enthält die Anfangsadresse des durchsuchten Feldes.
Eingang NX	Eingang NX enthält den Index zu dem Feld, bei dem die Suche beginnen soll.
IN	IN enthält das gesuchte Objekt.
Ausgang NX	Ausgang NX enthält die Position des Suchziels im Feld.
FD	FD zeigt an, dass ein Feldelement gefunden wurde und die Funktion erfolgreich war.
LEN	LEN gibt die Anzahl Elemente ab AR an, aus denen das durchsuchte Feld besteht. Der mögliche Bereich sind 1 bis 32.767 Byte oder Worte.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
AR		o	o	o	o	Δ	o	•	•	•		
NX Ein		•	•	•	•		•	•	•	•	•	
IN		o	o	o	o	Δ	o	•	•	•	•	
NX Aus		•	•	•	•		•	•	•	•		
FD	•											•

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Zulässige Referenz nur für INT, BYTE oder WORD, nicht zulässig für DINT.
- Δ Zulässige Referenz nur für BYTE oder WORD, nicht zulässig für INT oder DINT.

Beispiel 1

Das Feld AR ist definiert als Speicheradressen %R0001 — %R0005. Ist EN EIN, wird der Teil des Feldes zwischen %R0004 und %R0005 nach einem Element durchsucht, dessen Wert gleich IN ist. Sind %R0001 = 7, %R0002 = 9, %R0003 = 6, %R0004 = 7, %R0005 = 7 und %R0100 = 7, beginnt die Suche bei %R0004 und endet bei %R0004, wenn FD aktiviert wird. In %R0101 wird eine 4 eingetragen.

```
|          _____
|%I0001   |     |
|——| |————|SRCH_|
|         |     |
|         | EQ_ |
|         |     |
|         | INT |                                                      %Q0001
| %R0001—|AR FD|——————————————————————————————————————————————————————————( )—
|         | LEN |
|         |00005|
| CONST —|NX NX|— %R0101
| 00003   |     |
|         |     |
| %R0100—|IN    |
|         |_____|
|
```

Beispiel 2

Das Feld AR ist definiert als Speicheradressen %AI0001 — %AI0016. Die Werte dieser Feldelemente sind 100, 20, 0, 5, 90, 200, 0, 79, 102, 80, 24, 34, 987, 8, 0 und 500. Zunächst steht in %AQ0001 der Wert 5. Ist EN aktiviert, wird das Feld bei jedem Zyklus nach einer Übereinstimmung mit dem Wert 0 von IN durchsucht. Im ersten Zyklus beginnt die Suche bei %AI0006. Eine Übereinstimmung wird gefunden bei %AI0007. Hierauf wird FD durchgeschaltet und %AQ0001 auf 7 gesetzt. Im zweiten Zyklus beginnt die Suche bei %AI0008. Eine Übereinstimmung wird gefunden in %AI0015. FD bleibt somit EIN und %AQ0001 wird auf 15 gesetzt. Der nächste Zyklus beginnt bei %AI0016. Da jetzt das Feldende ohne Übereinstimmung erreicht wird, geht FD auf AUS und %AQ0001 wird auf Null gesetzt. Im nächsten Zyklus beginnt die Suche wieder am Anfang des Feldes.



Kapitel 11 Konvertierungsfunktionen

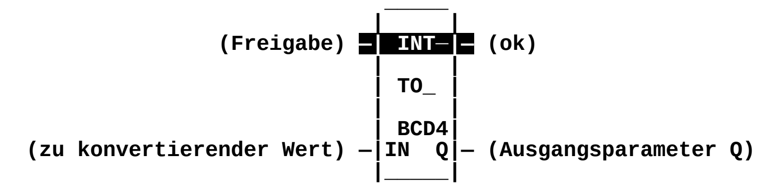
Mit den Konvertierungsfunktionen können Sie das Format eines Datenelements umwandeln. Zahlreiche Programmanweisungen, wie z.B. die arithmetischen Funktionen, verlangen die Verwendung eines bestimmten Datentyps. In diesem Kapitel werden folgende Konvertierungsfunktionen beschrieben:

Abkürzung	Funktion	Beschreibung	Seite
BCD-4	Umwandlung in BCD-4	Konvertiert eine vorzeichenbehaftete ganze Zahl in 4-stelliges BCD-Format.	11-2
INT	Umwandlung in ganze Zahl mit Vorzeichen	Konvertiert eine Zahl im BCD-4- oder REAL-Format in eine vorzeichenbehaftete ganze Zahl.	11-3
DINT	Umwandlung in doppeltgenaue ganze Zahl mit Vorzeichen	Konvertiert eine Zahl im REAL-Format in eine doppeltgenaue ganze Zahl mit Vorzeichen.	11-5
REAL	Umwandlung in REAL	Konvertiert eine Zahl im Format INT, DINT, BCD-4 oder WORD in das REAL-Format.	11-7
WORD	Umwandlung in WORD	Konvertiert eine Zahl im REAL-Format in das WORD-Format.	11-9
TRUN	Abschneiden	Rundet eine reelle Zahl gegen Null hin.	11-11

—>BCD-4 (INT)

Die BCD-4-Funktion gibt das vierstellige BCD-Äquivalent einer vorzeichenbehafteten ganzen Zahl aus. Die ursprünglichen Daten werden dabei nicht verändert. Die Ausgangsdaten können für den Betrieb BCD-kodierter LED-Anzeigen oder zur Voreinstellung externer Geräte (z.B. schnelle Zähler) verwendet werden.

Erhält die Funktion Stromfluss, führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange das Ergebnis der Konvertierung im Bereich zwischen 0 und 9999 liegt schaltet die Funktion den Stromfluss durch, wenn sie aktiviert wird.



Parameter

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den ganzzahligen Wert, der in BCD-4 konvertiert werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde.
Q	Ausgang Q enthält das BCD-4-Äquivalent des in IN spezifizierten Wertes.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•		•	•	•	•	•	
OK	•											•
Q		•	•	•	•		•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Wird im folgenden Beispiel der Eingang %I0002 gesetzt und liegt kein Fehler vor, wird die an den Eingängen %I0017 bis %I0032 anliegende vorzeichenbehaftete ganze Zahl in vier BCD-Stellen umgewandelt. Das Ergebnis wird in den Speicheradressen %Q0033 bis %Q0048 abgelegt. Mit der Spule %Q01432 kann überprüft werden, ob die Konvertierung erfolgreich war.

```
|           _____
|%I0002    |     |                                                        %Q1432
|——| |—————| INT_|———————————————————————————————————————————————————————————( )—
|          | TO_ |
|          |     |
|          | BCD4|
| %I0017———|IN  Q|—%Q0033
|
```

—>INT (BCD-4, REAL)

Die INT-Funktion gibt das ganzzahlige Äquivalent eines REAL- oder 4-stelligen BCD-Wertes aus. Die ursprünglichen Daten werden dabei nicht verändert.

Hinweis

Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Erhält die Funktion Stromfluss, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die Daten innerhalb des zulässigen Bereichs liegen, schaltet die Funktion den Stromfluss immer durch, wenn sie aktiviert wird.

```
                                  _____
                                 |     |
                   (Freigabe) —| BCD |— (ok)
                                 |     |
                                 | TO_ |
                                 |     |
                                 | INT |
(zu konvertierender Wert) —|IN   Q|— (Ausgangsparameter Q)
                                 |_____|
```

Parameter

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den Wert (BCD-4, REAL oder Konstante), der in einen ganzzahligen Wert konvertiert werden soll.
OK	Solange die Daten eine Zahl sind und innerhalb des zulässigen Bereichs liegen, wird der OK-Ausgang durchgeschaltet, wenn der Freigabeeingang aktiviert wird.
Q	Ausgang Q enthält die ganzzahlige Form des ursprünglichen Wertes aus IN.

Zulässige Speichertypen

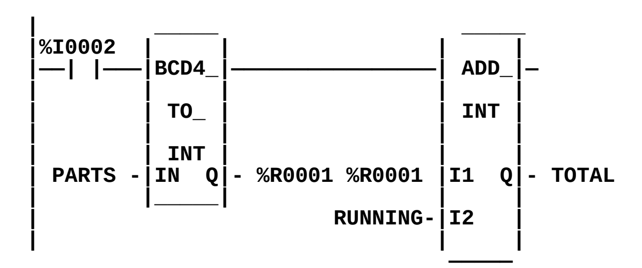
Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		•	•	•	•		•	•	•	•	•	
OK	•											•
Q		•	•	•	•		•	•	•	•		

Hinweis: Bei REAL-Daten sind nur die Typen %R, %AI und %AQ zulässig.

- Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Wird im folgenden Beispiel der Eingang %I0002 gesetzt, dann wird der vierstellige BCD-Wert in PARTS in eine vorzeichenbehaftete ganze Zahl konvertiert, die an die ADD-Funktion weitergegeben wird. Hier wird der Wert zu der durch die Referenz RUNNING spezifizierten vorzeichenbehafteten ganzen Zahl addiert. Das Ergebnis wird von der ADD-Funktion an die Referenz TOTAL ausgegeben.



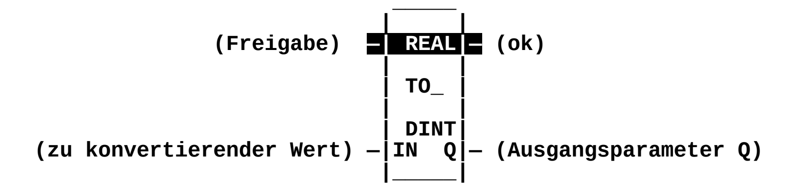
—>DINT (REAL)

Die DINT-Funktion gibt das vorzeichenbehaftete doppeltgenaue ganzzahlige Äquivalent eines Wertes im REAL-Format aus. Die ursprünglichen Daten werden dabei nicht verändert.

Hinweis

Den Datentyp REAL gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.

Erhält die Funktion Stromfluss, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange der REAL-Wert innerhalb des zulässigen Bereichs liegt gibt die Funktion immer Stromfluss weiter, wenn sie Stromfluss empfängt.



Parameter

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den Wert, der in einen doppeltgenauen ganzzahligen Wert konvertiert werden soll.
OK	Solange die REAL-Daten innerhalb des zulässigen Bereichs liegen, wird der OK-Ausgang durchgeschaltet, wenn der Freigabeeingang aktiviert wird.
Q	Ausgang Q enthält die vorzeichenbehaftete doppeltgenaue ganzzahlige Form des ursprünglichen Wertes aus IN.

Hinweis

Bei der Wandlung von REAL nach DINT kann Genauigkeit verlorengehen, da REAL 24 signifikante Bits besitzt.

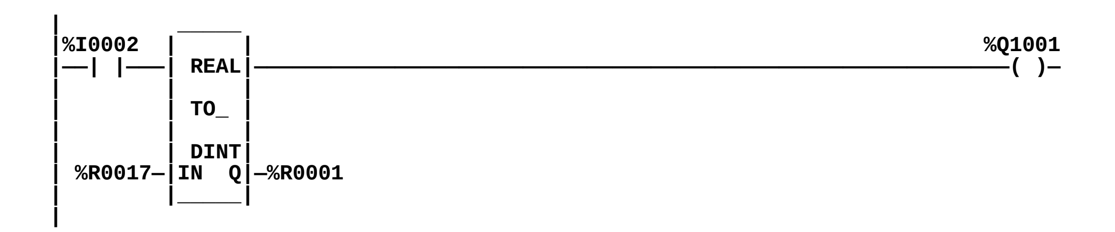
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•	
OK	•											•
Q								•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Wird im folgenden Beispiel der Eingang %I0002 gesetzt, dann wird der ganzzahlige Wert in der Eingangsadresse %I0017 in eine vorzeichenbehaftete doppeltgenaue ganze Zahl konvertiert. Das Ergebnis wird in die Adresse %R0001 eingetragen. Der Ausgang %Q1001 wird durchgeschaltet, wenn die Funktion erfolgreich ausgeführt wurde.



—>REAL (INT, DINT, BCD-4, WORD)

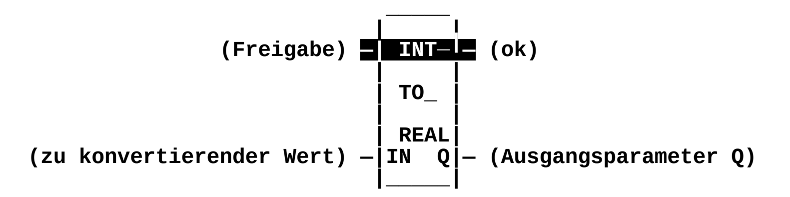
Diese Funktion gibt den reellen Wert der Eingangsdaten aus. Die ursprünglichen Daten werden dabei nicht verändert.

Erhält die Funktion Stromfluss, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die angegebene Konvertierung einen Wert innerhalb des zulässigen Bereichs ergibt, schaltet die Funktion den Stromfluss immer durch, wenn sie aktiviert wird.

Bei der Wandlung von DINT nach REAL kann Genauigkeit verloren gehen, da die Anzahl der signifikanten Bits auf 24 reduziert wird.

Hinweis

Diese Funktion gibt es nur bei den CPUs 35x und 36x, Ausgabestand 9 und höher, sowie bei allen Ausgabeständen der CPU352.



Parameter

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den ganzzahligen Wert, der in einen REAL-Wert konvertiert werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde.
Q	Ausgang Q enthält die REAL-Form des ursprünglichen Wertes aus IN.

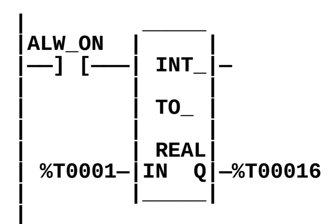
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN		o	o	o	o		o	•	•	•	•	
OK	•											•
Q								•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
o Nicht zulässig für DINT_TO_REAL.

Beispiel

Im folgenden Beispiel beträgt der ganzzahlige Wert des Eingangs IN 678. Das Ergebnis (678.000) wird in %T0016 abgelegt.



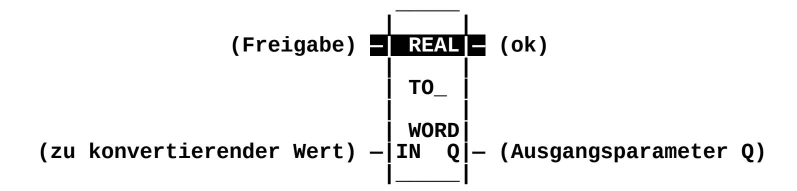
—>WORD (REAL)

Diese Funktion gibt das WORD-Äquivalent von REAL-Daten aus. Die ursprünglichen Daten werden dabei nicht verändert.

Hinweis

Diese Funktion gibt es nur bei den CPUs 35x und 36x.

Erhält die Funktion Stromfluss, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Solange die angegebene Konvertierung einen Wert innerhalb des zulässigen Bereichs (0 bis FFFFh) ergibt, schaltet die Funktion den Stromfluss immer durch, wenn sie aktiviert wird.



Parameter

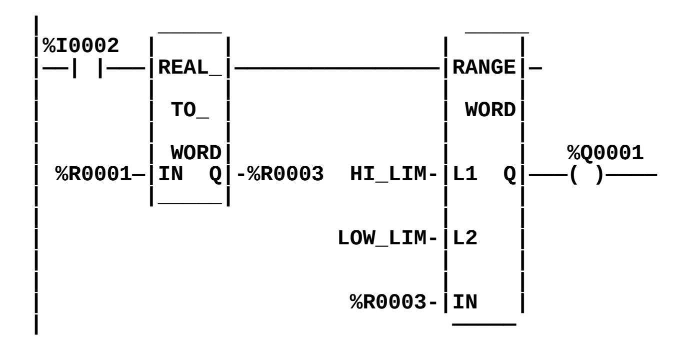
Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den Wert, der in das WORD-Format konvertiert werden soll.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde.
Q	Ausgang Q enthält die vorzeichenlose ganzzahlige Form des ursprünglichen Wertes aus IN.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN								•	•	•	•	
OK	•											•
Q		•	•	•	•		•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel


%I0002
REAL_
TO_
WORD
%R0001
IN
Q
%R0003
RANGE
WORD
HI_LIM
L1
Q
%Q0001
LOW_LIM
L2
%R0003
IN

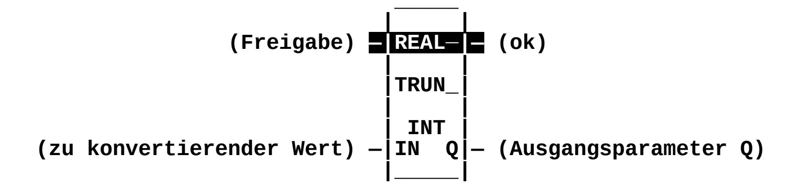
TRUN (INT, DINT)

Mit dieser Funktion kann ein reeller Wert gegen Null hin gerundet werden. Die ursprünglichen Daten werden dabei nicht verändert.

Hinweis

Die CPUs 35x und 36x (ab Ausgabestand 9 sowie alle Ausgabestände der CPU352) sind die einzigen CPUs der Series 90-30, die Gleitpunktarithmetik unterstützen. Die TRUN-Funktion hat daher für die übrigen CPUs der Series 90-30 keine Bedeutung.

Erhält die Funktion Stromfluss, dann führt sie die Konvertierung durch und stellt das Ergebnis am Ausgang Q bereit. Bei der CPU 352 schaltet die Funktion Stromfluss durch, wenn sie Stromfluss empfängt und die angegebene Konvertierung einen Wert ergibt, der eine Zahl ist und innerhalb des zulässigen Bereichs liegt. Bei allen anderen CPUs 35x und 36x gibt die Funktionen den Stromfluss *nicht* weiter.



Parameter

Parameter	Beschreibung
Freigabe	Die Konvertierung wird durchgeführt, wenn die Funktion freigegeben ist.
IN	IN enthält eine Referenz für den REAL-Wert, der abgeschnitten werden soll.
Ok	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wird, der Wert innerhalb des zulässigen Bereichs liegt und eine Zahl ist.
Q	Ausgang Q enthält den gekürzten INT- oder DINT-Wert des ursprünglichen Wertes aus IN.

Hinweis

Bei der Wandlung von REAL nach DINT kann Genauigkeit verlorengehen, da REAL 24 signifikante Bits besitzt.

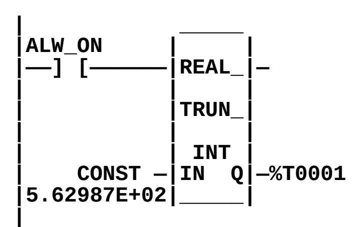
Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
IN								•	•	•	•	
OK	•											•
Q		o	o	o	o		o	•	•	•		

- • Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.
- o Nur für REAL_TRUN_INT zulässig.

Beispiel

Im folgenden Beispiel wird die angezeigte Konstante abgeschnitten und das ganzzahlige Ergebnis 562 wird in %T0001 eingetragen.



Kapitel 12

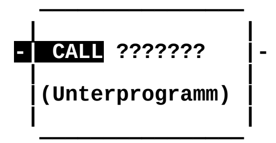
Steuerfunktionen

In diesem Kapitel werden die Steuerfunktionen beschrieben, mit denen die Programmausführung eingeschränkt und der von der CPU verfolgte Weg der Programmausführung abgeändert werden kann. Die Funktionsweise des CPU-Zyklus wird in Abschnitt 1 von Kapitel 2 erläutert.

Funktion	Beschreibung	Seite
CALL	Leitet die Programmbearbeitung auf einen Unterprogrammblock um.	12-2
DOIO	Aktualisiert während eines Zyklus einen bestimmten Bereich von Ein- oder Ausgängen unmittelbar. (Enthält eine DO I/O-Funktion Referenzadressen eines Moduls, werden sämtliche Ein- oder Ausgänge dieses Moduls aktualisiert, teilweise Aktualisierungen eines Moduls werden nicht durchgeführt). Wahlweise kann eine Kopie der bearbeiteten E/A im internen Speicher und nicht bei den echten Eingangspunkten abgelegt werden.	12-3
SER	Ereignisaufzeichnung — erfasst eine Reihe von Abtastwerten. Ein Funktionssteuerungsblock enthält eine externe Konfiguration von Funktionsblockablauf, Abtastwertkonfiguration und Betriebsparameter.	12-8
END	Bildet ein temporäres Programmende. Die Programmbearbeitung beginnt beim ersten Strompfad und endet entweder beim letzten Strompfad oder bei der END-Anweisung, was immer zuerst kommt. Diese Anweisung ist hilfreich beim Austesten eines Programms. Sie darf nicht bei SFC-Programmen eingesetzt werden (siehe Hinweis auf Seite 12-8).	12-21
MCR und MCRN	Programmierung einer Hauptsteuerrelaisfunktion. Die MCR-Funktion veranlasst, dass alle Strompfade zwischen MCR und dem nächsten ENDMCR ohne Stromfluss ausgeführt werden. Die Logicmaster 90-30/20/Micro Software unterstützt zwei MCR-Funktionstypen, die geschachtelte Form (MCRN) und die ungeschachtelte Form (MCR).	12-22
ENDMCR und ENDMCRN	Zeigt an, dass für die Ausführung der nachfolgenden Logik wieder Stromfluss erforderlich ist. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Funktionstypen, die geschachtelte (ENDMCRN) und die ungeschachtelte Form (ENDMCR).	12-25
JUMP und JUMPN	Bewirkt, dass bei der Programmausführung zu einer angegebenen Stelle (durch eine Marke gekennzeichnet; siehe LABEL) im Programm gesprungen wird. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Typen der JUMP-Funktion, die geschachtelte Form (JUMPN) und die ungeschachtelte Form (JUMP).	12-26
LABEL und LABELN	Gibt das Ziel einer JUMP-Anweisung an. Die Logicmaster 90-30/20/Micro Software unterstützt zwei Typen der LABEL-Funktion, die geschachtelte Form (LABELN) und die ungeschachtelte Form (LABEL).	12-28
COMMENT	Legt einen Kommentar (Strompfadkommentar) im Programm an. Nachdem die Anweisung programmiert wurde, kann der Text durch "Zoomen" der Anweisung eingegeben werden.	12-29
SVCREQ	Anforderung eines SPS-Spezialdienstes. (Eine Liste der Bedienanforderungen finden Sie auf Seite 12-30.)	12-30
PID	Es gibt zwei PID-Algorithmen: • Standard-ISA-PID-Algorithmus (PIDISA). • Unabhängiger PID-Algorithmus (PIDIND).	12-64

CALL

Mit der CALL-Funktion verzweigen Sie die Programmausführung zu einem in der Funktion angegebenen Unterprogrammblock.



Erhält die CALL-Funktion Stromfluss, verzweigt sie den Zyklus unmittelbar zum gewünschten Unterprogrammblock, der dann bearbeitet wird. Nach Abschluss des Unterprogrammblocks kehrt die Steuerung wieder zu dem Punkt zurück, der unmittelbar nach der CALL-Anweisung steht.

Beispiel

Das folgende Beispiel zeigt die CALL-Anweisung, wie sie im aufrufenden Block erscheint. Wenn Sie den Cursor auf die Anweisung setzen und dann **F10** drücken, zoomen Sie die Anzeige in das Unterprogramm.

```
|
|%I0004                                                              %T0001
|——| |—————————————————————————————————————————————————————————————————( )—
|
|               ________________
|              |                |
|%I0006        | CALL ASTRO     |-
|——| |————————| (Unterprogramm)|
|              |                |
|              |                |
|               ________________
|
|%I0003  %I0010                                                      %Q0010
|——| |——+——| |——————————————————————————————————————————————————————————( )—
|        |
|%I0001  |
|——| |——+
|
```

Hinweis

Die Micro SPS enthält keine Unterprogramme. Die CALL-Funktion kann daher mit der Micro SPS nicht benutzt werden.

DOIO

Mit der DO I/O-Funktion können Ein- und Ausgänge bei laufendem Programm für einen Zyklus aktualisiert werden. Die Funktion kann auch dazu benutzt werden, um bestimmte E/A zusätzlich zum normalen E/A-Zyklus während des Programms zu aktualisieren.

Werden Eingangsreferenzen angegeben, können durch die Funktion die neuesten Eingangswerte für das Programm erfasst werden. Werden Ausgangsreferenzen angegeben, aktualisiert DO I/O die Ausgänge auf der Grundlage der neuesten im E/A-Speicher abgelegten Werte. Bei der E/A werden immer komplette E/A-Module aktualisiert, falls erforderlich berichtigt die SPS während der Funktionsausführung die Referenzen.

Die DO I/O-Funktion besitzt vier Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluss und sind Eingangsreferenzen angegeben, werden die Eingangspunkte zwischen der Anfangsreferenz ST und END abgefragt. Wurde für ALT eine Referenz angegeben, wird eine Kopie des neuen Eingangswertes ab dieser Referenz im Speicher abgelegt, die realen Eingangspunkte werden nicht aktualisiert. ALT muss genauso groß sein wie der abgefragte Referenztyp. Wird für ST und END eine diskrete Referenz benutzt, muss ALT ebenfalls diskret sein. Wird für ALT keine Referenz angegeben, werden die realen Eingangspunkte aktualisiert.

Erhält die Funktion Stromfluss und sind Ausgangsreferenzen angegeben, werden die Ausgangspunkte zwischen der Anfangsreferenz ST und END zu den Ausgangsmodulen übertragen. Sollen Ausgangswerte von einem von %Q oder %AQ verschiedenen internen Speicher an die Ausgangsmodulen gegeben werden, kann die Anfangsreferenz für ALT angegeben werden. Der Ausgabebereich, der zu den Ausgangsmodulen übertragen wird, ist durch die Anfangsreferenz ST und die Endereferenz END bestimmt.

Die Funktion wird solange ausgeführt, bis entweder alle Eingänge des gewählten Bereiches abgefragt wurden oder alle Ausgänge auf den E/A-Modulen aktualisiert wurden. Die Programmausführung kehrt dann zur nächsten Funktion nach der DO I/O-Funktion zurück.

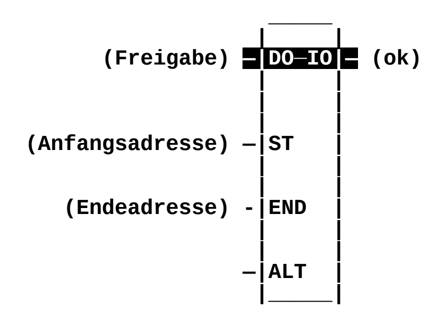
Enthält der Referenzbereich ein Zusatzmodul (HSC, APM, usw.), werden alle Ein- (%I und %AI) bzw. Ausgangsdaten (%Q und %AQ) für dieses Modul abgefragt. Bei der Abfrage von Zusatzmodulen wird der ALT-Parameter ignoriert. Im Referenzbereich darf kein erweitertes GCM-Modul liegen (siehe nachstehenden Hinweis).

Hinweis

> Bei CPUs ab Ausgabestand 9.0 ***kann*** die DOIO-Funktion zusammen mit dem erweiterten GCM-Modul verwendet werden.

Die Funktion gibt Stromfluss nach rechts weiter, wenn sie Stromfluss erhält, mit Ausnahme von:

- Nicht alle Referenzen des angegebenen Typs sind im gewählten Bereich vorhanden.
- Die CPU kann die von der Funktion angelegte temporäre E/A-Liste nicht ordnungsgemäß verarbeiten.
- Im Bereich befinden sich E/A-Module, die mit einem Fehler "E/A-Verlust" verknüpft sind.



Parameter

Parameter	Beschreibung
Freigabe	Bei Funktionsfreigabe wird ein begrenzter Ein- oder Ausgabezyklus durchgeführt.
ST	ST ist die Anfangsadresse eines Satzes von Ein- oder Ausgangspunkten oder Worten, die bedient werden sollen.
END	END ist die Endeadresse eines Satzes von Ein- oder Ausgangspunkten oder Worten, die bedient werden sollen.
ALT	Beim Eingabezyklus gibt ALT die Adresse an, in der die abgefragten Eingangspunkte/Worte gespeichert werden sollen. Beim Ausgabezyklus gibt ALT die Adresse an, von der die Ausgangspunkte/Worte abgeholt werden können, die zu den E/A-Modulen geschickt werden. Bei CPUs ab Modell 331 kann der ALT-Parameter eine Auswirkung auf die Ausführungsgeschwindigkeit des DOIO-Funktionsblockes haben (siehe nachstehenden Hinweis und Abschnitt über die erweiterte DOIO-Funktion für CPUs ab Modell 331 auf Seite 12-4).
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Ein- oder Ausgangszyklen normal ausgeführt wurden.

Hinweis

Bei CPUs ab Modell 331 aufwärts kann der ALT-Parameter der DOIO-Funktion dazu benutzt werden, den Steckplatz eines einzelnen Moduls im Hauptchassis einzugeben. Wird dies gemacht, wird der DOIO-Funktionsblock in 80 Mikrosekunden bearbeitet (anstatt 236 Mikrosekunden, wenn der Block ohne den ALT-Parameter programmiert wurde). Es findet keine Überprüfung zur Verhinderung von Adresskonflikten oder Moduldiskrepanzen statt.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
ST		•	•						•	•		
END		•	•						•	•		
ALT		•	•	•	•		•	•	•	•		•
OK	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Eingabebeispiel 1

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Referenzen %I0001 bis %I0064 abgefragt und %Q0001 wird durchgeschaltet. Eine Kopie der abgefragten Eingänge wird im internen Speicher ab Referenz %M0001 bis Referenz %M0064 abgelegt. Die echten Eingangspunkte werden nicht aktualisiert. Mit dieser Funktionsart können die aktuellen Werte der Eingangspunkte mit den Werten der Eingangspunkte am Anfang des Zyklus verglichen werden.

```
|
|           _____
|%I0001    |     |                                                        %Q0001
|--| |-----|DO_IO|--------------------------------------------------------( )-
|          |     |
|          |     |
|%I0001 - |ST    |
|          |     |
|          |     |
|%I0064 - |END   |
|          |     |
|          |     |
|%M0001 --|ALT   |
|          |_____|
|
```

Eingabebeispiel 2

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Referenzen %I0001 bis %I0064 abgefragt und %Q0001 wird durchgeschaltet. Die abgefragten Eingänge werden im Eingangszustandsspeicher ab Referenz %I0001 bis Referenz %I0064 abgelegt. Mit dieser Funktionsart können Eingangspunkte während des Programmausführungsteils eines CPU-Zyklus mehrmals abgefragt werden.

```
|
|           _____
|%I0001    |     |                                                        %Q0001
|--| |-----|DO_IO|--------------------------------------------------------( )-
|          |     |
|          |     |
|%I0001 - |ST    |
|          |     |
|          |     |
|%I0064 - |END   |
|          |     |
|          |     |
|         -|ALT  |
|          |_____|
|
```

Ausgabebeispiel 1

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Werte der analogen Ausgangskanäle %AQ001 bis %AQ004 in die Referenzen %R0001 bis %R0004 eingetragen und %Q0001 wird durchgeschaltet. Die Werte aus %AQ001 bis %AQ004 werden nicht zu den analogen Ausgangsmodulen übertragen.

```
|
|             _____
|%I0001      |     |                                                       %Q0001
|--| |-------|DO_IO|-------------------------------------------------------( )-
|            |     |
|            |     |
|%AQ001 -    |ST   |
|            |     |
|            |     |
|%AQ004 -    |END  |
|            |     |
|            |     |
|%R0001 -    |ALT  |
|            |_____|
|
```

Ausgabebeispiel 2

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann werden die Werte aus den Referenzen %AQ001 bis %AQ004 in die analogen Ausgangskanäle %AQ001 bis %AQ004 eingetragen und %Q0001 wird durchgeschaltet.

```
|
|             _____
|%I0001      |     |                                                       %Q0001
|--| |-------|DO_IO|-------------------------------------------------------( )-
|            |     |
|            |     |
|%AQ001 -    |ST   |
|            |     |
|            |     |
|%AQ004 -    |END  |
|            |     |
|            |     |
|           -|ALT  |
|            |_____|
|
```

Erweiterte DOIO-Funktion für CPUs ab Modell 331 aufwärts

Achtung

Wird die erweiterte DOIO-Funktion in einem Programm verwendet, darf dieses Programm nicht mit einer Version der Logicmaster 90-30/20 Software geladen werden, die älter als 4.01 ist.

Die erweiterte Version der DOIO-Funktion (E/A-Aktualisierung) ist verfügbar für CPUs ab Modell 331 und Ausgabestände ab 4.20. Die erweiterte Version der DOIO-Funktion kann nur auf ein einziges diskretes Ein- oder Ausgangsmodul mit 8, 16 oder 32 Punkten angewandt werden.

Mit dem ALT-Parameter wird der Steckplatz des Moduls im Hauptchassis identifiziert. Eine Konstante 2 in diesem Parameter gibt der CPU zum Beispiel an, dass die erweiterte Version des DOIO-Funktionsblocks auf das in Steckplatz 2 eingebaute Modul angewandt werden soll.

Hinweis

Die erweiterte DOIO-Funktion überprüft nur, ob das Modul in dem angegebenen Steckplatz funktionstüchtig ist.

Die erweiterte DOIO-Funktion kann nur für Module im Hauptchassis verwendet werden. Der Wert des ALT-Parameters muss daher zwischen 2 und 5 (Chassis mit 5 Steckplätzen) bzw. 2 und 10 (Chassis mit 10 Steckplätzen) liegen.

Anfangs- und Endereferenzen müssen vom Typ %I oder %Q sein. Diese Referenzen geben die erste und letze Referenz an, für die das Modul konfiguriert wurde. Wurde zum Beispiel ein Eingangsmodul mit 16 Punkten für %I0001 bis %I0016 in Steckplatz 10 eines Chassis mit 10 Steckplätzen konfiguriert, muss der ST-Parameter %I0001 sein, der END- Parameter %I0016 und de ALT-Parameter 10 (siehe nachstehende Abbildung).

```
|
|           _____
|%I0001    |     |                                                  %Q0001
|--| |-----|DO_IO|--------------------------------------------------( )-
|          |     |
|          |     |
|%I0001 - |ST    |
|          |     |
|          |     |
|%I0016 - |END   |
|          |     |
|          |     |
|     10 --|ALT  |
|          |_____|
|
```

Die Tabelle vergleicht die Ausführungszeiten eines normalen DOIO-Funktionsblocks für ein E/A-Modul mit 8, 16 oder 32 Punkten mit denen eines erweiterten DOIO-Funktionsblocks.

Modul	Ausführungszeit normale DOIO	Ausführungszeit erweiterte DOIO
Diskretes Eingangsmodul mit 8 Punkten Diskretes Ausgangsmodul mit 8 Punkten	224 Mikrosekunden 208 Mikrosekunden	67 Mikrosekunden 48 Mikrosekunden
Diskretes Eingangsmodul mit 16 Punkten Diskretes Ausgangsmodul mit 16 Punkten	224 Mikrosekunden 211 Mikrosekunden	68 Mikrosekunden 47 Mikrosekunden
Diskretes Eingangsmodul mit 32 Punkten Diskretes Ausgangsmodul mit 32 Punkten	247 Mikrosekunden 226 Mikrosekunden	91 Mikrosekunden 50 Mikrosekunden

SER

Eigenschaften

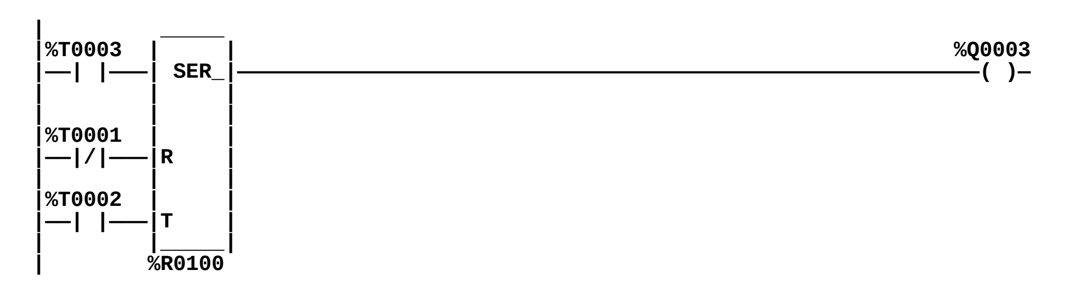
- Der SER-Funktionsblock (Ereignisaufzeichnung) erfasst eine Reihe von Abtastwerten. Erhält der Freigabeeingang des SER-Funktionsblocks Stromfluss, dann erfasst der Funktionsblock pro Abtastung 32 zusammenhängende oder nicht zusammenhängende Bits.
- Jeder SER-Funktionsblock kann bis zu 1024-Abtastwerte erfassen.
- Wird der SER-Funktionsblock in ein periodisch abgearbeitetes Unterprogramm eingebettet, dann wird die Abtastrate durch die Ausführungsrate dieses Unterprogramms bestimmt.
- Nur der Triggerabtastwert erhält einen Zeitstempel. Der Triggerabtastwert kann mit einem Zeitstempel im BCD- (die maximale Auflösung beträgt 1 Sekunde) oder POSIX-Format (maximale Auflösung 10 ms) versehen werden. Der Zeitstempel wird nur einmal am Triggerpunkt platziert. Die SER-Funktion unterstützt nur einen Zeitstempel pro Aufzeichnung.
- Die SER-Funktion kann so konfiguriert werden, dass die Erfassung vor, während oder nach dem Trigger erfolgt (siehe Seite 12-14.)
- Die SER-Operation wird durch einen Funktionsteuerblock konfiguriert, den Sie mit einer Reihe von Blockverschiebungsbefehlen (BLKMV) erstellen können (siehe Seite 12-10).

Hinweis

Die Synchronisation von SPS zu SPS wird nicht unterstützt.

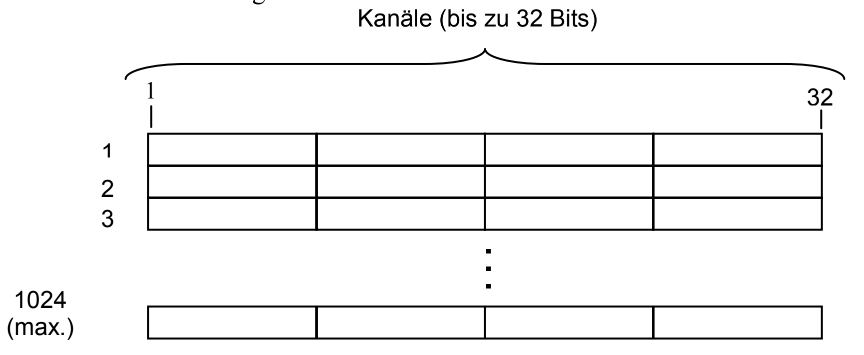
Der Funktionsblock besitzt einen Aus- und drei Eingänge: Freigabe, Rücksetzen und Trigger.

SER-Funktionsblock



Hinweis

Diese Funktion verlangt CPU-Firmware ab Ausgabestand 9.00. Sie ist ***nur*** in CPUs ab 350 verfügbar.



Parameter

Parameter	Beschreibung
Freigabe	Bei jeder Freigabe und ausgeschaltetem Reset der Funktion erfasst der SER-Funktionsblock einen Abtastwert aller konfigurierten Kanäle.
R	Empfängt der Rücksetzeingang Stromfluss, dann wird die SER-Funktion unabhängig vom Zustand des Freigabeeingangs zurückgesetzt. Abtastwertpuffer, Triggerabtastwert-offset, Triggerzeit und aktueller Abtastwertoffset werden alle auf Null gesetzt. Der Funktionsblock bleibt solange im Rücksetzzustand, bis der Stromfluss vom Rücksetz-eingang weg genommen wird. Im Rücksetzzustand wird der OK-Ausgang abgeschaltet. Die Abtastung wird wieder aufgenommen, sobald der Stromfluss vom Rücksetzeingang weg genommen wird.
T	Wird der Funktionsblock bei aktivem Triggereingangsmodus freigegeben, dann schaltet der SER-Funktionsblock in den getriggerten Zustand, wenn der Triggereingang einge-schaltet wird. Triggerzeit, Triggerabtastwertoffset und ein Abtastwert werden aufge-zeichnet. Der Triggerabtastwert wird unabhängig von der Anzahl erfasster Abtastwerte aufge-zeichnet. Nachdem er einmal angestoßen wurde, fährt der Ereignisschreiber solange mit der Erfassung von Abtastwerten fort, bis die "Anzahl Abtastwerte nach dem Trigger" erreicht ist. Jetzt hält er mit der Erfassung von Abtastwerten solange an, bis am Rück-setzeingang Stromfluss anliegt. Das Triggersignal wird ignoriert, wenn der Triggermodus auf "voller Puffer" eingestellt wird. Informationen zur Konfiguration des Triggermodus finden Sie unter "Funktionssteue-rungsblock" auf Seite 12-10.
Anfangsrefe-renz	Das aus 78 Worten bestehende Funktionssteuerungsblockfeld beginnt bei dieser Refe-renz. Der Funktionssteuerungsblock legt die Ausführung des Funktionsblocks, die Ab-tastwertkonfiguration und die Betriebsparameter fest. Einzelheiten finden Sie unter "Funktionssteuerungsblock" auf Seite 12-10.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Triggerbedingungen (die durch den Parameter "Triggermodus" festgelegt sind) erfüllt sind und die gesamte Abtastung abge-schlossen ist. Unabhängig vom Zustand des Freigabeeingangs erhält der Ausgang solan-ge Stromfluss, bis Stromfluss am Rücksetzeingang anliegt.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
Steue-rungsblock								•				
R	•											
T	•											
OK	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Funktionssteuerungsblock

Der Funktionssteuerungsblock ist ein 78-Wort-Feld, das Datenerfassung und Triggermechanismen für die SER-Funktion definiert. In einem Programm kann jeweils nur ein SER-Funktionsblock mit jedem Funktionsbefehlsblock und Datenblock verknüpft werden.

Konfigurieren Sie die Parameter des SER-Funktionsblocks in folgenden Schritten:

1. Stellen Sie die gespeicherten Werte für das Feld entsprechend nachstehender Tabelle ein. Sie können entweder die Register mit Blockverschiebungen initialisieren oder die Daten in der Registertabelle initialisieren und die Tabelle vor Aktivierung der SER-Funktion speichern.
2. Fügen Sie den SER-Funktionsblock in Ihr Kontaktplanprogramm ein.

Hinweis

Wenn Sie x Kanäle benötigen, und x ein Wert ist, der ungleich 8, 16 oder 24, aber kleiner als 32 ist, dann müssen Sie eine Anzahl Kanäle auswählen, die größer als x und ein Vielfaches von 8 ist, und für die nicht benutzten Kanäle eine Nullkanalbeschreibung einfügen. Eine Nullkanalbeschreibung besitzt einen Segmentselektor von 0xFFh, einen Längenparameter, der gleich der Anzahl nicht benutzter Kanäle ist, sowie einen Offset von 0.

Wort	Parameter	Beschreibung
0 (Anfangsreferenz)	Status	Variable, die den aktuellen Zustand des SER-Funktionsblocks angibt. Diese Variable kann nur gelesen werden. Weitere Informationen finden Sie in den Statuszusatzdaten (Wort 1). **Hinweis:** Wird im Steuerungsblock ein Fehler erkannt, dann wird der Status auf 6 gesetzt, der OK-Ausgang wird abgeschaltet und es erfolgt keine Aktion. Zu den Statuseinstellungen gehören: 0 = Rücksetzen 1 = Inaktiv 2 = Aktiv 3 = Getriggert 4 = Abgeschlossen 5 = Überlauffehler 6 = Parameterfehler
1	Statuszusatzdaten	Diese Variable, die nur gelesen werden kann, liefert weitere Zustandsinformationen über die SER-Funktion. Die Einstellung dieses Parameters finden Sie unter "Statuszusatzdaten-Zustände" auf Seite 12-12.
2	Triggermodus	Definiert die Bedingungen, unter denen der SER-Funktionsblock in den getriggerten Zustand übergeht. Zulässige Einstellungen sind: 0 = Modus "Triggereingang" 1 = Modus "voller Puffer" Wird der Funktionsblock im Modus "Triggereingang" freigegeben, dann wird bei Aktivierung des Triggersignals ein Zeitstempel erzeugt. Die Abtastung wird solange fortgesetzt, bis die "Anzahl Abtastwerte nach Trigger" erfüllt ist. Ist dies der Fall, wird der OK-Ausgang aktiviert. Im Modus "voller Puffer" wird das Triggersignal ignoriert. Ist der Funktionsblock freigegeben, wird die Abtastung solange fortgesetzt, bis der Abtastwertpuffer voll ist. Ist dies der Fall, wird der OK-Ausgang aktiviert. Die Puffergröße wird mit dem Parameter "Anzahl Abtastwerte" eingestellt.

Wort	Para-meter	Beschreibung
3	Trigger-zeit-format	Hiermit wird festgelegt, wie die Triggerzeit angezeigt wird. Für BCD-Anzeige wird dieser Parameter auf 0 gesetzt, für POSIX-Anzeige auf 1 (Einzelheiten siehe Seite 12-20.)
4—7	Reser-viert	Die Worte 4 bis 7 sind reserviert und müssen auf Null gesetzt werden.
8	Anzahl Kanäle (Bits pro Abtast-wert)	Dieser Parameter gibt an, wieviel Datenbits bei jeder Ausführung des Funktionsblocks erfasst und an den Abtastwertpuffer zurück gegeben werden. Mögliche Werte sind 8, 16, 24 oder 32 Bits. Die Werte sind bytestrukturiert (8 Bits). Unbenutzte Kanäle müssen mit einer Nullkanalbeschreibung konfiguriert werden (siehe Worte 14—77.)
9	Anzahl Abtast-werte	Gibt die Größe des Abtastwertpuffers an. Zulässige Werte sind 1 bis 1024 Abtastwerte (die tatsächliche Buffergröße in Bits ergibt sich aus der Anzahl Abtastwerte mal der Anzahl Kanäle).
10	Anzahl Abtast-werte nach Trigger	Dieser Parameter gibt an, wieviel Abtastwerte erfasst werden, nachdem die Triggerbedin-gung wahr wurde. Der Parameter kann auf einen Wert zwischen Null und (Anzahl Abtast-werte – 1) gesetzt werden. Er ist nur gültig, wenn der Triggermodus auf "Triggereingang" (0) gesetzt wurde.
11	Ein-gangs-modul-Steck-platz	Dieser Parameter gibt an, wo sich das Eingangsmodul für die Datenerfassung befindet (Steckplatz im Hauptchassis). Das Eingangsmodul wird nicht abgefragt, wenn dieser Para-meter auf 0 gesetzt ist. Beim Abfragen eines Eingangsmoduls werden dessen Werte lokal gespeichert. Die Werte der für das Modul konfigurierten Referenzadressen sind nicht be-troffen. Um Werte von den abgefragten Eingangsmodulen im Datenblock-Abtastwertpuffer zu speichern ist eine Kanalbeschreibung erforderlich. Die zurückgeschickten Daten sind Null, wenn das Modul zum Zeitpunkt der Abtastung nicht vorhanden oder gestört ist. In diesem Fall wird kein Fehler in der Fehlertabelle eingetragen. Die Fehleranzeige wird der E/A-Zyklussteuerung überlassen.
12	Da-tenblock seg-mentse-lektor	Dieser Parameter legt den Datentyp des Datenblocks fest. Wollen Sie zum Beispiel bei %R0100 beginnen, müssen Sie bei diesem Parameter den Wert 08 eintragen. Zulässige Werte für diesen Parameter sind unter anderem: %R (08h), %AI (0Ah), %AQ (0Ch). Ein-zelheiten zum Datenblock finden Sie auf Seite 12-13.
13	Datenbl o-ckoffset	Dieser Parameter gibt die Anfangsreferenz für den Datenblock an. Er ist absolut. Wollen Sie zum Beispiel bei %R0100 beginnen, müssen Sie bei diesem Parameter den Wert 99 eintragen. Stellen Sie sicher, dass genügend Speicherplatz für den gesamten Datenblock vorhanden ist.
14—77	Kanalbe schrei-bungen	Dieser Parameter gibt die mit einem bestimmten Kanal verknüpfte Referenzadresse (Seg-mentselektor, Länge und Offset) an. Je nach Anzahl abgetasteter Kanäle und Datenlänge kann es zwischen 1 und 32 Kanalbeschreibungen geben. Die Daten werden in der in diesem Abschnitt festgelegten Reihenfolge zurück gegeben.
	Kanal-seg-mentse-lektor/ Länge	Dieses Wort, das als Hexadezimalwert eingegeben wird, definiert Segmentselektor und Datenlänge (in Bits). MSB = Segmentselektor. LSB = Datenlänge. Die Datenlänge ist bei zusammenhängenden Abtastwerten hilfreich. Der Segmentselektor kann auf jeden diskreten Datentyp eingestellt werden: %I (46h), %Q (48h), %M (4Ch), %T (4Ah), %G (56h), %S (54h), %SA (4Eh), %SB (50h), %SC (52h), Nullselektor (FFh) und Eingangsmodulselektor (00h).
		Der Längenparameter kann im Bereich zwischen 1 und 32 liegen. Die Summe aller Längen darf jedoch nicht größer sein als der Parameter "Anzahl Kanäle". Bei einer Länge größer 1 können mehrere zusammenhängende Kanäle mit einer einzigen Kanalbeschreibung konfi-guriert werden.
	Kanal-offset	Als Hexadezimalwert eingegeben definiert dieses Wort den Bit-Offset des im Segmentse-lektor angegebenen Datentyps oder Eingangsmoduls. Dieser Offset ist absolut. Der Bereich dieses Parameters verändert sich mit dem Segmentselektor (Datentyp und Länge). Der Offset gibt an, an welcher Stelle in Datentabelle oder Eingangsmodul die Abtastung statt-finden soll.

Status der Statuszusatzdaten

Die Statuszusatzdaten (Wort 1 im Funktionssteuerungsblock) liefern zusätzliche Zustandsinformationen für die SER-Funktion.

Wert	Zustand	Beschreibung
0	Rücksetz-zustand	Erhält der Rücksetzeingang Stromfluss, werden Abtastwertpuffer, Triggerabtastwertoffset, Triggerzeit und aktueller Abtastwertoffset alle auf Null gesetzt. Der Ausgang wird abgeschaltet gehalten. Wird der Stromfluss vom Rücksetzeingang weg genommen, erfolgt ein Übergang zum ***inaktiven Zustand***. Die Statuszusatzdaten besitzen keine Bedeutung und werden gelöscht.
1	Inaktiv	Zustand zwischen Rücksetzzustand und aktivem Zustand. In diesem Zustand werden keine Aktionen durchgeführt. Der SER-Ausgang wird abgeschaltet gehalten. Der Übergang zum aktiven Zustand tritt auf, wenn der Funktionsblock am Freigabeeingang Stromfluss erhält.
2	Aktiv	Am Freigabeeingang liegt Stromfluss, aber der Funktionsblock ist nicht rückgesetzt, fehlerhaft oder getriggert. Bei freigegebenem Funktionsblock wird bei jeder Ausführung ein Abtastwert aufgezeichnet. Der Ausgang wird abgeschaltet gehalten. Die Triggerbedingung (durch den Triggermodusparameter festgelegt) wird überwacht und verursacht einen Wechsel in den getriggerten Zustand, wenn die Bedingungen wahr werden. Wurde mehr als die "Anzahl Abtastwerte" aufgenommen, dann werden die Statuszusatzdaten auf 0x01 gesetzt. Im anderen Fall sind sie 0x00.
3	Getriggert	In diesem Zustand ist die durch den Triggermodus festgelegte Triggerbedingung wahr. Je nach Triggermodus und Parametereinstellungen werden zusätzliche Abtastwerte genommen. Der Ausgang wird abgeschaltet gehalten. Sind alle Abfragen abgeschlossen, findet ein Wechsel in den Zustand "Beendet" statt. Wurde mehr als die "Anzahl Abtastwerte" aufgenommen, dann werden die Statuszusatzdaten auf 0x01 gesetzt. Im anderen Fall sind sie 0x00.
4	Beendet	Alle Abfragen sind abgeschlossen. Der Ausgang erhält Stromfluss. Es ist nur der Übergang in den Rücksetzzustand erlaubt. Wurde mehr als die "Anzahl Abtastwerte" aufgenommen, dann werden die Statuszusatzdaten auf 0x01 gesetzt. Im anderen Fall sind sie 0x00.
5	Überlauf-fehler	Der Steuerungs-/Datenblock geht über das Ende seines Speichertyps hinaus. Der Ausgang wird abgeschaltet gehalten. Es ist nur der Übergang in den Rücksetzzustand erlaubt. Die Statuszusatzdaten besitzen keine Bedeutung und werden gelöscht.
6	Parameter-fehler	Im Funktionssteuerungsblock oder anderen Betriebsparametern liegt ein Fehler vor. Der Ausgang wird abgeschaltet gehalten. Es ist nur der Übergang in den Rücksetzzustand erlaubt. Das Statuszusatzdatenwort enthält den Offset zu dem Steuerungsblock, bei dem der Parameterfehler aufgetreten ist.
7	Statusfeh-ler	Der Statusparameter ist ungültig. Der Ausgang wird abgeschaltet gehalten. Es ist nur der Übergang in den Rücksetzzustand erlaubt. Der ungültige Statuswert wird in der Statuszusatzdatenadresse im Steuerungsblock gespeichert.

SER-Datenblockformat

Der SER-Datenblock enthält Abtastwertpuffer, Abtastwertoffsets und Triggerdaten. Diese Informationen werden von der CPU geliefert. Sie sollten diesen Datenbereich daher nur lesen. Es liegt in Ihrer Verantwortung, dem Datenblock ausreichend Registerbereich zur Verfügung zu stellen. Das Blockformat ist wie folgt:

Wort*	Parameterbeschreibung
0	Nummer des aktuellen Abtastwertoffset. Spricht die Adresse an, bei der der jüngste Abtastwert abgelegt wurde. Der Parameter ist absolut. Der zulässige Bereich ist –1 bis 1023. Registeradresse von Abtastwert = (Anzahl Byte pro Abtastwert) * (Offsetparameter)/2 + (Abtastwertpuffer-Anfangsregister). **Hinweis:** Dieser Wert wird erst gültig, nachdem ein Abtastwert erfasst wurde. Der Wert wird auf –1 gesetzt, wenn die SER-Funktion über den Rücksetzeingang rückgesetzt wurde.
1	Nummer des Triggerabtastwertoffsets. Spricht die Speicheradresse des Abtastwertes an, der beim Übergang der Triggerbedingung in den Zustand "wahr" erfasst wurde. Der Parameter ist absolut. Der zulässige Bereich ist 0 bis 1023. Registeradresse von Abtastwert = (Anzahl Byte pro Abtastwert) * (Offsetparameter)/2 + (Abtastwertpuffer-Anfangsregister). **Hinweis:** Dieser Wert wird erst gültig, nachdem die Triggerbedingung erfüllt wurde. Der Wert wird auf 0 gesetzt, wenn die SER-Funktion über den Rücksetzeingang zurückgesetzt wurde.
2 bis 5	Triggerzeit: Dieser Wert gibt den Zeitpunkt (entsprechend der Zeituhr in der SPS) an, zu dem die Triggerbedingung im Funktionsblock wahr wurde. Der Zeitwert kann im BCD-Format (Voreinstellung) oder POSIX-Format angezeigt werden. Das Format wird vom Parameter *Triggerzeitformat* im Steuerungsblock festgelegt. Bei Aktivierung des Rücksetzeingangs wird dieser Wert auf den Anfangswert Null gesetzt.
6 bis Ende des Abtastwert-puffers.	Abtastwertpuffer. Der Speicherbereich, der die Abtastwerte enthält. Bei Aktivierung des Rücksetzparameters wird dieser Bereich auf Null gesetzt. Die Größe des Abtastwertpuffers ist veränderlich. Sie hängt von den Anzahl Kanäle und der Größe der Abtastwerte ab. Der Abtastwertpuffer ist ein Ringpuffer – nachdem die letzte Adresse vollgeschrieben wurde, überschreibt der nächste Abtastwert den Abtastwert im ersten Register. Ende des Abtastwertpuffers = 5 + ({[(Anzahl zu erfassender Abtastwerte) * (Anzahl abzutastender Kanäle / 8)] +1} / 2

*Offset von der durch Datenblocksegmentselektor (Wort 12) und Datenblock festgelegten Anfangsreferenz Offset (Wort 13) im Funktionssteuerungsblock.

SER-Arbeitsweise

Wird der SER-Funktionsblock während der Abfrage freigegeben, liest er die konfigurierten Abtastpunkte und schreibt sie in einen Ringpuffer. Der Ausgang wird durchgeschaltet, nachdem die konfigurierte Anzahl Abtastwerte aufgenommen wurde. Der Wechsel des Ausgangszustands kann dazu verwendet werden, die seit Erfassung des letzten Abtastwerts verstrichene Zeit zu erfassen oder zusätzliche Abfragen anzustoßen (siehe "Abfragemodi”).

Vor Abfrage muss der SER- Funktionsblock rückgesetzt werden (Freigabe des Stromflusses zum Rücksetzeingang). Durch das Rücksetzen wird der Datenblockbereich initialisiert. Wird er niccht rückgesetzt, wird der Funktionsblockstatus mit den aktuellen Werten im Datenblock ausgeführt. Hierdurch wird der aktuelle Abtastwertoffset falsch und die Daten im Datenblock sind ungültig.

Bei jeder Ausführung des Funktionsblocks im rückgesetzten, aktiven oder getriggerten Zustand wird der Steuerungsblock des SER-Funktionsblocks abgefragt. Wird ein Konfigurationsparameter

im Steuerungsblock während der Programmbearbeitung verändert, dann wird diese Änderung bei der nächsten Abfrage des mit diesem Steuerungsblock verknüpften SER-Funktionsblocks wirksam. Tritt ein Fehler auf, wird die Operation angehalten und der Funktionsblock wechselt in den entsprechenden Fehlerzustand. Um mit der Abfrage fortzufahren müssen Sie den Fehler beheben und dann den Funktionsblock rücksetzen (Freigabe des Stromflusses zum Rücksetzeingang).

Wählen Sie ein Eingangsmodul für die Abfrage aus, prüft die SPS *nicht*, ob es sich um ein diskretes Eingangsmodul handelt oder ob die mit dem Modul verknüpften Kanalbeschreibungen entsprechend der Modulgröße zulässige Längen- und Offsetwerte besitzen. Sie müssen die Abfrage eines Eingangsmoduls korrekt einstellen. Obwohl sich mehrere Kanalbeschreibungen an ein Eingangsmodul richten können, wird das Modul pro Funktionsblockausführung nur einmal abgefragt.

Der SER-Funktionsblock kann im normalen Anwenderprogramm oder in ein periodisch ablaufendes Unterprogramm eingesetzt werden. Im Anwenderprogramm entspricht die Auflösung des Intervalls zwischen den Abfragen der Auflösung der Zykluszeit, die entsprechend Anzahl und Art der bei den einzelnen Zyklen aktiven Funktionen unterschiedlich sein kann. In einem Interrupt-Unterprogramm kann das Zeitintervall bis auf 1 ms herunter gehen. Die Auflösung ist hier mit wenig Schwankung auf 1 ms reproduzierbar.

Die Bearbeitungszeit eines Funktionsblocks mit einem periodisch ablaufenden 1-ms-Unterprogramm kann bis zu 50% der CPU-Ressourcen verbrauchen. Sie sollten in einem periodisch ablaufenden 1-ms-Unterprogramm nicht mehr als zwei SER-Funktionen einplanen.

Abfragemodi

Der SER-Abfragemodus wird durch die Parameter "Triggermodus" (Wort 2 im Funktionssteuerungsblock) und "Anzahl Abtastwerte nach Trigger" (Wort 10) bestimmt. Sie müssen den Inhalt des Abtastwertpuffers auf der Grundlage der Konfiguration dieser Parameter interpretieren.

Triggergesteuerte Abfrage

Zur Konfiguration der Abfragemodi vor, während oder nach dem Trigger wird der Triggermodus (Wort 2 = 0) eingestellt. Der Abfragemodus wird durch die Anzahl Abtastwerte nach Trigger (Wort 10) gesteuert. In allen Fällen beginnt die Abfrage damit, dass das Freigabesignal auf H-Pegel geht. Geht das Triggersignal auf H-Pegel, wird die Abfrage solange fortgesetzt, bis die Anzahl Abtastwerte nach Trigger erfasst wurde. Das Funktionsblock-Ausgangssignal geht auf H-Pegel, wenn die Abfrage abgeschlossen ist.

Wird mehr als die konfigurierte "Anzahl Abtastwerte" (Wort 9) erfasst, ehe die Bedingung "Anzahl Abtastwerte nach Trigger" erfüllt ist, geht die SER-Funktion wieder an den Anfang des Puffers zurück und überschreibt die zuvor erfassten Abtastwerte.

Wenn der Trigger zum ersten Mal von AUS nach EIN umschaltet, wird die Triggerzeit unter einer konfigurierten Adresse abgelegt.

"Vor Trigger"

Erfasst solange kontinuierlich Abtastwerte, bis das Triggersignal erkannt wird.

Zur Konfiguration dieses Modus setzen Sie Wort 10 auf Null. Wenn das Triggersignal aktiviert wird hält die Erfassung an und es wird ein Zeitstempel erzeugt. (Alle Abtastwerte wurden vor dem Triggersignal erfasst.)

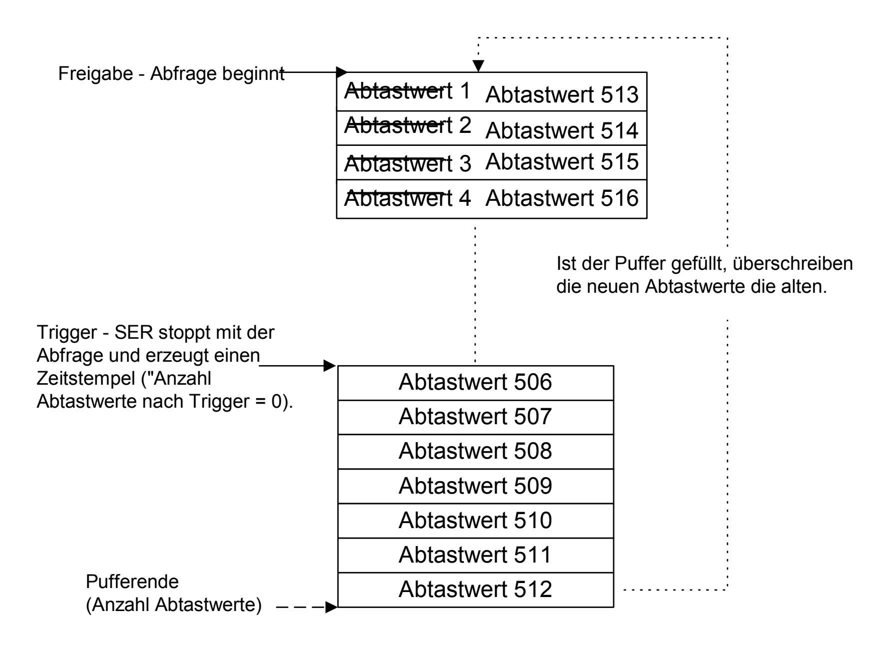


Abbildung 12-1. Beispiel einer SER-Abfrage vor dem Trigger

"Während Trigger"

Es werden solange kontinuierlich Abtastwerte erfasst, bis die "Anzahl Abtastwerte nach Trigger" erfasst wurde.

Zur Konfiguration dieses Modus müssen Sie Wort 10 auf einen Wert zwischen 1 und der Anzahl Abtastwerte (Wort 9) setzen. Wird das Triggersignal aktiviert, wird die Abfrage solange fortgesetzt, bis die konfigurierte Anzahl Werte erfasst wurde. Im nachstehenden Beispiel hat "Anzahl Abtastwerte nach Trigger" den Wert 12. Ist die Abfrage abgeschlossen, enthält der Puffer 500 Abtastwerte vor Trigger und 12 Abtastwerte nach Trigger.

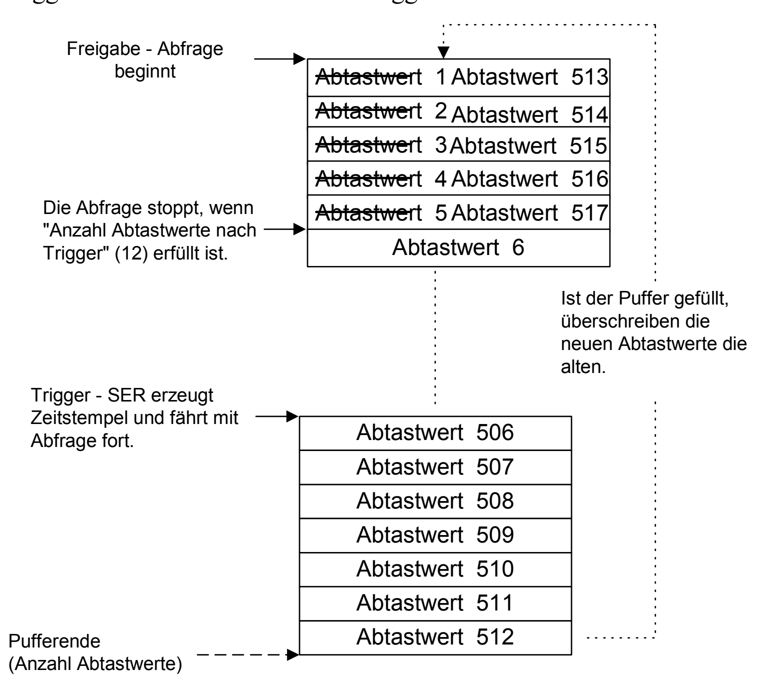


Abbildung 12-2. Beispiel einer SER-Abfrage während des Triggers

Nach Trigger

Erfasst solange kontinuierlich Abtastwerte, bis die Anzahl Abtastwerte erreicht ist.

Zur Konfiguration dieses Modus müssen Sie Wort 10 auf einen Wert setzen, der gleich der Anzahl Abtastwerte (Wort 9) ist. Wird das Triggersignal aktiviert, wird die Abfrage solange fortgesetzt, bis die konfigurierte Anzahl Werte erfasst wurde. (Alle Abtastwerte nach dem Triggersignal.)

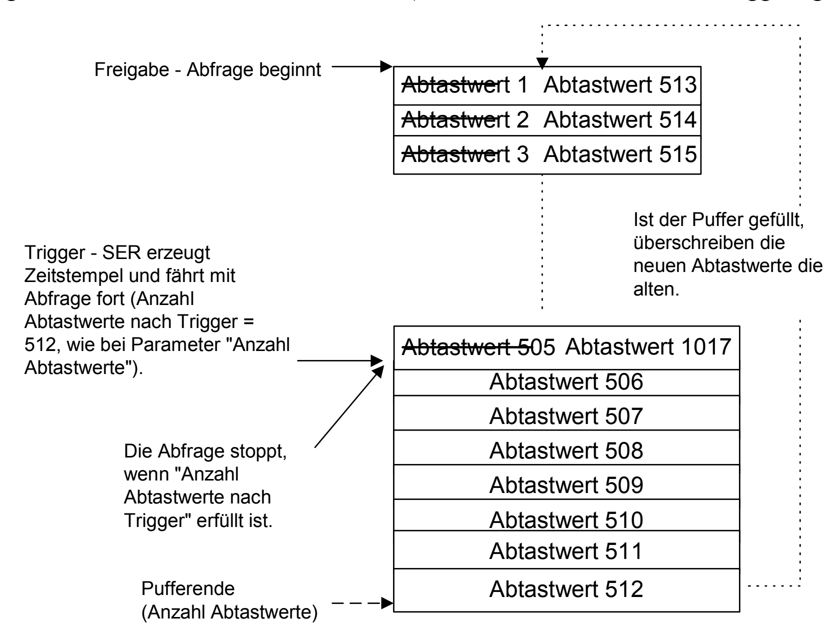


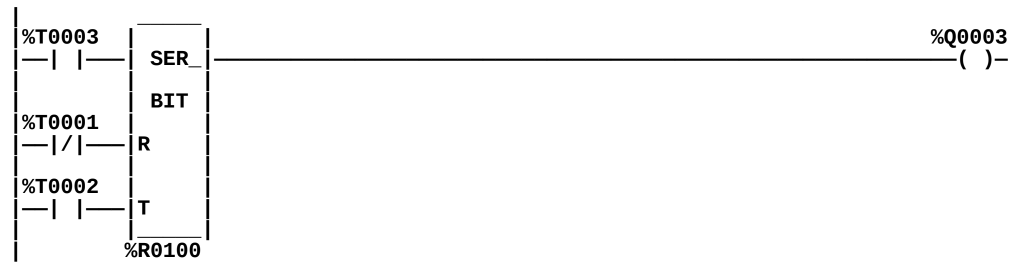
Abbildung 12-3. SER-Erfassung nach Trigger

Voller Puffer (Trigger kontrolliert nicht die Abfrage)

Wird Triggermodus auf 1 eingestellt, dann wird der Parameter "Anzahl Abtastwerte nach Trigger" (Wort 10) ignoriert und das Triggereingangssignal hat keine Wirkung auf die Arbeitsweise des Funktionsblocks. Ist der Funktionsblock aktiviert, wird die Abfrage solange fortgesetzt, bis die Anzahl Abtastwerte (Wort 8) erfasst ist. Hierdurch wird der Abtastwertpuffer gefüllt. Ist der Puffer voll, hält die Abfrage an, ein Triggerzeitstempel wird erzeugt und der OK-Ausgang des Funktionsblocks geht auf H-Pegel.

Beispiel einer SER-Funktion

Im folgenden Beispiel wurde der Funktionssteuerungsblock entsprechend der Beschreibung in Tabelle 12-1 eingestellt.



Beispiel eines Funktionssteuerungsblocks

Das System in diesem Beispiel besitzt in Steckplatz 4 von Chassis 0 ein diskretes Eingangsmodul mit 16 Punkten. Es wurde solange ausgeführt, bis 572 Abtastwerte (512 + 60) genommen wurden. Am Freigabeeingang liegt Stromfluss an, an den Rücksetz- und Triggereingängen nicht.

Tabelle 12-1. Funktionssteuerungsblock für SER-Beispiel

Wort	Register	Parameter	Wert (dez.)	Wert (hexa)	Beschreibung
0	%R0100	Status	2	0002	Der Funktionsblock ist aktiv. Dies bedeutet, dass der Funktionsblock normal abläuft und jedes Mal, wenn er im Programmablauf angetroffen wird, einen Abtastwert erfasst.
1	101	Statuszusatzdaten	1	0001	Die Statuszusatzdaten geben an, dass mehr als 512 Abtastwerte erfasst wurden und die Befüllung des Abtastwertpuffers bereits mindestens einmal wieder von vorne begonnen hat.
2	102	Triggermodus	0	0000	Der Ereignisschreiber ist so konfiguriert, dass er auf der Grundlage des Triggereingangs anläuft.
3	103	Triggerzeitformat	0	0000	BCD
4	104	Reserviert	0	0000	Die "Reserviert" Parameter werden immer auf 0 gesetzt.
5	105	Reserviert	0	0000	
6	106	Reserviert	0	0000	
7	107	Reserviert	0	0000	
8	108	Anzahl Kanäle	24	0018	Die Abtastwertkonfiguration besteht aus 24 Datenbits.
9	109	Anzahl zu erfassender Abtastwerte	512	0200	Die Größe des Abtastwertpuffers beträgt 512 Abtastwerte. Beachten Sie, dass der Abtastwertpuffer nicht 512 Byte groß ist. Seine Größe beträgt 512 x (24/8) = 1536 Byte oder 768 Worte (jeder Abtastwert ist 3 Byte lang).
10	110	Anzahl Abtastwerte nach Trigger	12	000C	Die Anzahl der Abtastwerte, die erfasst wird, nachdem der Trigger 12 ist.
11	111	Eingangsmodul-Steckplatz	4	0004	Das Eingangsmodul in Steckplatz 4 von Chassis 0 wird erfasst, so dass seine Istwerte für die Abfrage zur Verfügung stehen.
12	112	Datenblocksegmentselektor	8	0008	Das Datensegment ist 0x08 (%R).
13	113	Datenblockoffset	200	00C8	Der Offset ist 200. Hierdurch wird der Anfang des Datenblocks auf %R0201 gelegt. Der Offset ist zwar ein absoluter Wert, die Registertabellen beginnen aber bei %R0001. Daher ergibt sich der Anfangspunkt des Datenblocks zu %R0001 + 200 = %R0201.

Wort	Register	Parameter	Wert (dez.)	Wert (hexa)	Beschreibung
Wort	**Register**	**Parameter**	**Wert (dez.)**	**Wert (hexa)**	**Beschreibung**
Kanalbeschreibungen		Die restlichen Worte enthalten die Kanalbeschreibungen. In diesem Beispiel wurden sechs Kanalbeschreibungen definiert.			
14	114	Seg.-Einst. : Länge	17921	4601	Kanalbeschreibung 1: Mit der ersten Kanalbeschreibung wird das %I-Segment mit einer Länge von 1 und einem Offset von 0 eingestellt. Hierdurch wird %I0001 für Kanal 1 gewählt.
15	115	Offset	0	0000	
16	116	Seg.-Einst. : Länge	-253	FF03	Kanalbeschreibung 2: Mit der zweiten Kanalbeschreibung wird der Nullselektor mit einer Länge von 3 und einem Offset von 0 eingestellt. Der Nullselektor bewirkt, das die Kanäle 2 - 4 ignoriert oder "übersprungen" werden. Diese Kanäle enthalten immer einen Abtastwert von Null.
17	117	Offset	0	0000	
18	118	Seg.-Einst. : Länge	3	0003	Kanalbeschreibung 3: Mit der dritten Kanalbeschreibung wird der Eingangsmodulselektor mit einer Länge von 3 und einem Offset von 12 eingestellt. Der Eingangsmodulselektor veranlasst, dass vom Eingangsmodul Abtastwerte genommen werden. Diese Kanalbeschreibung wählt die Werte in den Punkten 13, 14 und 15 des Eingangsmoduls für die Kanäle 5 - 7 aus.
19	119	Offset	12	0012	
20	120	Seg.-Einst. : Länge	18434	4802	Kanalbeschreibung 4: Mit der vierten Kanalbeschreibung wird das %Q-Segment mit einer Länge von 2 und einem Offset von 8 eingestellt. Hierdurch werden %Q0009 und %Q0010 für die Kanäle 8 und 9 ausgewählt.
21	121	Offset	8	0008	
22	122	Seg.-Einst. : Länge	8	0008	Kanalbeschreibung 5: Die fünfte Kanalbeschreibung ist ein weiterer Eingangsmodulselektor. Er besitzt eine Länge von 8 und einen Offset von 0. Hierdurch werden die Werte für die Punkte 1 bis 8 des Moduls in die Kanäle 10 - 17 platziert.
23	123	Offset	0	0000	
24	124	Seg. Einst. : Länge	-249	FF07	Kanalbeschreibung 6: Die sechste Kanalbeschreibung ist ein weiterer Nullselektor. Er besitzt eine Länge von 7 und einen Offset von 0. Diese Nullkanalbeschreibung bewirkt, dass die Kanäle 18 - 24 mit Nullen gefüllt werden. Diese letzte Kanalbeschreibung wird benötigt, um den Abtastwertpuffer bis zu den im Parameter "Anzahl Kanäle" angegebenen 24 Bits aufzufüllen. Da alle 24 Kanäle konfiguriert sind, werden keine weiteren Kanalbeschreibungen mehr benötigt.
25	125	Offset	0	0000	

Abtastwert-Inhalt

Tabelle 12-2 fasst die in einem einzelnen Abtastwert enthaltenen Werte auf der Grundlage der Kanalbeschreibungen im Abtastwert-Steuerungsblock zusammen.

Tabelle 12-2. Abtastwert-Inhalt für SER-Beispiel

Kanalnummer	Kanalinhalt
1	%I0001
2 - 4	Nullen
5	Eingangsmodul-Punkt 13
6	Eingangsmodul-Punkt 14
7	Eingangsmodul-Punkt 15
8	%Q0009
9	%Q0010
10 - 17	Eingangsmodul-Punkte 1 - 8
18 - 24	Nullen

Datenblock für Steuerungsblockbeispiel

In Tabelle 12-3 wird das Datenblockformat aufgelistet, das sich aus dem auf Seite 12-17 vorgestellten Steuerungsblockbeispiel ergibt. Beachten Sie, dass es bei Register 201 beginnt, wie dies von den Segmentoffsetparametern (Worte 12 und 13) im Steuerungsblock beschrieben wird.

Tabelle 12-3. Datenblock für SER-Steuerungsblockbeispiel

Offset	Register	Parameterbeschreibung	Wert (dez.)	Wert (hexa)
0	%R0201	Nummer des aktuellen Abtastwertoffset	59	003B
1	202	Nummer des Triggerabtastwertoffset	0	0000
2 - 5	203 – 206	Triggerzeit (BCD)	0 0 0 0	0000 0000 0000 0000
6 - 768	207 – 975	Abtastwertpuffer	Abtastwertdaten	Abtastwertdaten

Der aktuelle Abtastwertoffset ist 59. Das bedeutet, dass der 59. Abtastwert der letzte Abtastwert ist, der in den Abtastwertpuffer eingetragen wird (nicht 59 Register). Bei 3 Byte pro Abtastwert ist der aktuelle Offset bei 59 * 3 = 177 Byte (oder dem oberen Byte des 89. Registers). Da die Triggerbedingungen nicht erfüllt wurden, sind Triggerabtastwert und Triggerzeit gleich 0 und der Ausgang wird nicht durchgeschaltet. Der Abtastwertpuffer enthält 512 Abtastwerte, von denen 59 der neueste und 60 der älteste ist.

SER-Funktionsblock Trigger-Zeitstempelformate

Beispiel: Triggerzeit 3. November 1998 um 8:34:05:16 Uhr

BCD-Format:

```
struct time_of_day_clk_rec {
        unsigned char  Sekunden;
        unsigned char  Minuten;
        unsigned char  Stunden;
        unsigned char  Tag;
        unsigned char  Monat;
        unsigned char  Jahr;
};
```

Register	Parameter	Wert (dez.)	Wert (hexa)
%R203	Minuten/Sekunden	13317	3405
%R204	Tagesdatum/Stunden	776	0308
%R205	Jahr/Monat	-26607	9811
%R206	nicht benutzt	0	0

POSIX-Format:

```
struct timespec {
        long    tv_sec; /* Anzahl Sekunden seit 1. Januar 1970 */
        long    tv_nsec;/* Anzahl Nanosekunden in die nächste Sekunde
                           hinein */
};
```

Register	Parameter	Wert (dez.)	Wert (hexa)
%R0203	Zweitniedrigstes Wort	-7811	e17d
%R204	Zweithöchstes Wort	13845	3615
%R205	Nanosekunden unteres Wort	26624	6800
%R206	Nanosekunden oberes Wort	2441	0989

END

Die END-Funktion ermöglicht ein temporäres Programmende. Die Programmbearbeitung beginnt beim ersten Strompfad und endet entweder beim letzten Strompfad oder bei der END-Anweisung, was immer zuerst kommt.

Die END-Funktion beendet die Programmausführung unbedingt. Nach der END-Funktion darf im Strompfad nichts mehr stehen. Nach der END-Funktion wird keine Logik mehr bearbeitet und die Programmsteuerung wird für den nächsten Zyklus an den Programmanfang übergeben.

Die END-Funktion ist hilfreich beim Austesten, da sie die Ausführung der nach ihr stehenden Logik verhindert.

Bei der Logicmaster-Programmiersoftware zeigt eine Marke [END OF PROGRAM LOGIC] (Programmende) das Ende der Programmausführung an. Diese Marke wird verwendet, wenn im Programm keine END-Funktion programmiert wurde.



Beispiel

Im folgenden Beispiel wurde ein END programmiert, um den aktuellen Zyklus zu beenden.

```
|
|  STOP
|
|-[  END  ]
|
```

Hinweis

Bei CPUs ab Ausgabestand 7 entsteht ein Fehler "END Function Executed from SFC Action" [Ende-Funktion aus SFC-Aktion ausgeführt], wenn Sie eine END-Funktion in SFC oder in ein von SFC aufgerufenes Programm platzieren. (Bei CPUs bis zum Ausgabestand 7 gab es hierbei Probleme im Ablauf, aber keine Fehlermeldung.) Weitere Angaben zu diesem Fehler finden Sie unter "Diskrepanz bei Systemkonfiguration" in Abschnitt 2 von Kapitel 3.

MCRN/MCR

Ein Hauptsteuerrelais (MCR) muss immer zusammen mit einem Hauptsteuerrelais-Ende (ENDMCR) verwendet werden. Beide Funktionen müssen den gleichen Namen haben. Sämtliche Strompfade zwischen einer aktiven Hauptsteuerrelaisfunktion MCR und dem zugehörigen ENDMCR werden ohne Stromfluss zu den Spulen ausgeführt. Nach der zu MCR gehörenden Funktion ENDMCR wird die normale Programmausführung wieder aufgenommen. Im Gegensatz zu der JUMP-Funktion können MCR-Funktionen nur in Vorwärtsrichtung verwendet werden. Die Anweisung ENDMCR muss in der Programmabfolge nach der MCR-Funktion auftreten.

Die MCR-Funktion hat folgende Auswirkungen:

- Zeitgeber werden nicht verändert. TMR-Typen werden rückgesetzt. Beim ONDTR-Funktionsblock hält der Akkumulator seinen Wert.
- Normale Ausgänge sind AUS, negierte Ausgänge sind EIN.

Hinweis

Wird eine MCR-Funktion durchgeschaltet, wird zwar die von ihr gesteuerte Logik bearbeitet und der Kontaktstatus angezeigt, aber es werden keine Ausgänge durchgeschaltet. Wenn Sie nicht merken, dass die beobachtete Logik von einer MCR-Funktion kontrolliert wird, kann es wie eine Störung aussehen. Die Software markiert daher auf dem Bildschirm den von der MCR-Funktion kontrollierten Programmbereich mit einer doppelten Stromschiene.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der MCR-Funktion, eine geschachtelte und eine ungeschachtelte Form.

CPU-Kompatibilität

CPU-Typ	Vergleich mit Maske
CPUs Serie 35x und 36x (ab Ausgabestand 2)	Nur geschachtelte Form verwenden (MCRN)
Series 90 CPUs, Ausgabestand 1	Nur nicht geschachtelte Form verwenden (MCR)

MCRN-Operation

Solange eine MCRN-Funktion im Verhältnis zu anderen MCRN-Funktionen ordnungsgemäß geschachtelt wurde und nicht im Wirkungsbereich ungeschachtelter MCR- und JUMP-Funktionen liegt, kann sie an beliebiger Stelle im Programm angeordnet sein.

Eine MCRN-Anweisung und die zugehörige ENDMCRN-Anweisung müssen vollständig innerhalb eines anderen MCRN/ENDMCRN-Paares liegen. Es sind acht Schachtelungsebenen möglich. Ein Beispiel finden Sie auf Seite 12-24.

Hinweis

Verwenden Sie bei den CPUs 35x und 36x für jedes ENDMCRN nur ein MCRN.

Mit Ausnahme der zuvor erwähnten CPUs 35x und 36x können zu einem einzelnen ENDMCRN mehrere MCRN-Funktionen gehören. Dies entspricht der geschachtelten JUMP-Funktion, bei der mehrere Sprünge auf das gleiche Sprungziel (LABEL) gerichtet sein können. Die Unterschiede zwischen der Sprungfunktion und der MCR-Funktion werden im Abschnitt "Unterschiede zwischen MCR und JUMP” auf Seite 12-23 beschrieben.

MCR-Operation

Zu jeder ENDMCR-Anweisung kann nur eine MCR-Anweisung gehören. Der Bereich einer ungeschachtelten MCR/ENDMCR-Funktion darf sich nicht mit dem Bereich eines anderen MCR/ENDMCR-oder JUMP/LABEL-Paares überschneiden oder einen solchen Bereich enthalten. Eine ungeschachtelte MCR-Funktion darf nicht im Wirkungsbereich eines JUMP/LABEL-Paares liegen.

Parameter

Beide Formen der MCR-Funktion besitzen die gleichen Parameter. Sie besitzen beide einen Booleschen Freigabeeingang (EN) sowie eine Bezeichnung, über die sie identifiziert werden. Diese Bezeichnung wird am Ende mit der ENDMCR-Anweisung wiederholt. Weder MCR noch MCRN besitzen Ausgänge, daher ist nach einer MCR-Funktion keine weitere Funktion im Strompfad mehr möglich.

```
 ???????                  ???????
  [ MCR ]       oder      -[ MCRN]
```

Unterschiede zwischen MCR und JUMP

Bei der MCR-Funktion werden Funktionsblöcke im Wirkungsbereich der MCR-Funktion ***ohne Stromfluss*** ausgeführt und Spulen ***werden abgeschaltet***. Ist im nachstehenden Beispiel %I0002 EIN, wird die MCR-Funktion freigegeben. Bei freigegebener MCR-Funktion wird — selbst wenn %I0001 EIN ist — der ADD-Funktionsblock ausgeführt, ***ohne*** dass Strom fließt (d.h. er addiert keine 1 zu %R0001), und %Q0001 wird abgeschaltet.

```
|
|%I0002   FIRST
|——| |———[ MCR ]
|
|
|             _____
|%I0001      |     |                                      %Q0001
|——| |———————| ADD |————————————————————————————————————————( )-
|            | INT |
|            |     |
|     %R0001-|I1  Q|— %R0001
|            |     |
|            |     |
|          1—|I2   |
|            |_____|
|
+[   ENDMCR   ]
```

Bei der JUMP-Funktion werden die Funktionsblöcke zwischen JUMP und LABEL ***nicht*** ausgeführt und die Spulen ***sind nicht betroffen***. Im nachstehenden Beispiel wird der Sprung ausgeführt, wenn %I0002 EIN ist. Da der Programmteil zwischen JUMP und LABEL übersprungen wird, wird %Q0001 nicht beeinflusst (d.h. er bleibt EIN, wenn er zuvor EIN war, bzw. AUS, wenn er AUS war).

```
|%I0001
|——| |————————————————————————————————————————————————————————————>>TEST1
|                _____
|%I0001         |     |                                           %Q0001
|——| |—————————| ADD |——————————————————————————————————————————————( )-
|               | INT |
|               |     |
|      %R0001-|I1  Q|- %R0001
|               |     |
|               |     |
|           1-|I2    |
|               |_____|
|
| --TEST1 :
```

Beispiel

Das folgende Beispiel zeigt eine MCRN mit der Bezeichnung "SECOND", die in einer MCRN mit der Bezeichnung "FIRST" geschachtelt ist. Wird über %I0002 die MCRN-Funktion aktiviert, dann wird das Programm ohne Stromfluss zu den Spulen fortgesetzt, bis die zugehörige Funktion ENDMCRN erreicht ist. Sind %I0001 und %I0003 EIN, geht %Q0001 AUS und %Q0003 bleibt EIN.

Zur Hilfe bei der Fehlersuche in Kontaktplanprogrammen wird der von einer MCR-Funktion kontrollierte Programmabschnitt durch eine doppelte Stromschiene gekennzeichnet.

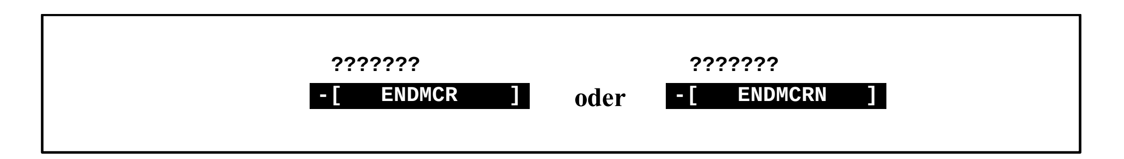
```
 |
 |%I0002   FIRST
 |——| |———[ MCRN ]
 |
||
||
||%I0004   SECOND
||——| |———[ MCRN ]
||
||
||
||%I0001                                                          %Q0001
||——| |—————————————————————————————————————————————————————————————( )—
||
||
||
||%I0003                                                          %Q0003
||——| |—————————————————————————————————————————————————————————————(S)—
||
||
|| SECOND
|+[   ENDMCRN   ]
||
||
||
 | FIRST
 +[   ENDMCRN   ]
 |
```

ENDMCRN/ENDMCR

Mit ENDMCR (Ende Hauptsteuerrelais) wird nach einer MCR-Funktion wieder der normale Programmablauf aufgenommen. Nach einer aktiven MCR-Funktion veranlasst die zugehörige ENDMCR-Funktion, dass das Programm wieder mit normalem Stromfluss fortgesetzt wird. War die zugehörige MCR-Funktion nicht aktiv, dann hat die ENDMCR-Funktion keine Wirkung.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der ENDMCR-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form (ENDMCR) muss mit der ungeschachtelten MCR-Funktion (MCR) verwendet werden. Die geschachtelte Form (ENDMCRN) muss mit der geschachtelten MCR-Funktion (MCRN) verwendet werden.

Die ENDMCR-Funktion besitzt einen negierten Booleschen Eingang (EN). Die Funktionsfreigabe muss von der Stromschiene kommen, die Ausführung kann nicht bedingt erfolgen. Die ENDMCR-Funktion besitzt auch eine Bezeichnung, mit der sie identifiziert und der (den) entsprechenden MCR(s) zugeordnet wird. Sie besitzt keine Ausgänge, daher ist nach einer ENDMCR-Anweisung keine weitere Funktion im Strompfad mehr möglich.



Beispiel

Die ENDMCR-Anweisung im folgenden Beispiel beendet den Bereich "CLEAR" der MCR-Funktion.

Beispiel einer nicht geschachtelten ENDMCR-Funktion:

```
|
|  CLEAR
|
|-[   ENDMCR    ]
|
```

Beispiel einer geschachtelten ENDMCR-Funktion:

```
|
|  CLEAR
|
|-[   ENDMCRN   ]
|
```

JUMP

Mit der JUMP-Funktion kann ein Teil des Programms übersprungen werden. Die Programmausführung wird an der durch LABEL markierten Stelle fortgesetzt. Ist JUMP aktiv, werden alle Spulen innerhalb des Wirkungsbereiches der Funktion eingefroren. Hierzu gehören auch Merker, die mit Zeitgebern, Zählern, Haftrelais und Relais verknüpft sind.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der JUMP-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form, die seit Software-Ausgabestand 1 verfügbar ist, hat die Form ——————>>LABEL01, wobei LABEL01 die Bezeichnung der entsprechenden ungeschachtelten LABEL-Anweisung ist.

Bei ungeschachtelten JUMP-Funktionen kann immer nur eine JUMP-Anweisung einer LABEL-Anweisung zugeordnet werden. Der Sprung kann entweder nach vorne oder nach hinten ausgeführt werden.

Der Bereich einer ungeschachtelten JUMP/LABEL-Funktion darf sich nicht mit dem Bereich eines anderen MCR/ENDMCR-oder JUMP/LABEL-Paares überschneiden. Eine ungeschachtelte JUMP-Funktion sowie die zugehörige LABEL-Funktion dürfen nicht im Wirkungsbereich eines anderen MCR/ENDMCR- oder JUMP/LABEL-Paares liegen. Darüber hinaus darf ein MCR/ENDMCR-Paar oder ein anderes JUMP/LABEL-Paar nicht im Wirkungsbereich eines ungeschachtelten JUMP/LABEL-Paares liegen.

Hinweis

Die ungeschachtelte JUMP-Funktion ist die einzige Sprungfunktion, die in Ausgabe 1 der SPS Series 90-30 verwendet werden kann. Die geschachtelte JUMP-Funktion kann (und sollte) für alle neuen Anwendungen verwendet werden.

Beachten Sie auch, dass die CPUs 35x und 36x nur geschachtelte Sprünge unterstützen, keine ungeschachtelten Sprünge.

Die geschachtelte Form der JUMP-Anweisung hat die Form ———N———>>LABEL01, wobei LABEL01 die Bezeichnung der entsprechenden geschachtelten LABEL-Anweisung ist. Die geschachtelte Form der JUMP-Funktion ist ab Ausgabestand 2 der Logicmaster 90-30/20/Micro Software und SPS-Firmware verfügbar.

Solange eine geschachtelte JUMP-Anweisung nicht im Wirkungsbereich einer ungeschachtelten MCR- oder JUMP-Funktion liegt, darf sie an beliebiger Stelle im Programm angelegt werden.

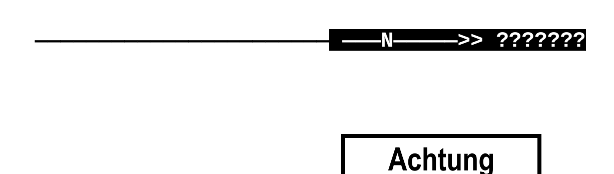
Zu einer geschachtelten LABEL-Anweisung können mehrere geschachtelte JUMP-Funktionen gehören. Geschachtelte JUMP-Funktionen können vorwärts oder rückwärts ausgeführt werden.

Beide JUMP-Funktionen werden immer in die Spalten 9 und 10 der aktuellen Strompfadzeile eingetragen. Nach einer JUMP-Funktion ist keine weitere Funktion im Strompfad mehr möglich. Der Stromfluss erfolgt direkt von der Anweisung zu dem Strompfad mit der angegebenen Bezeichnung.

Ungeschachtelte JUMP-Funktion:



Geschachtelte JUMP-Funktion:



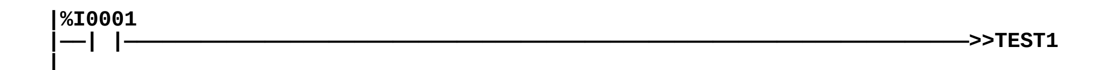
Achtung

Um Endlosschleifen zu vermeiden, muss ein Rückwärtssprung immer eine Bedingung enthalten.

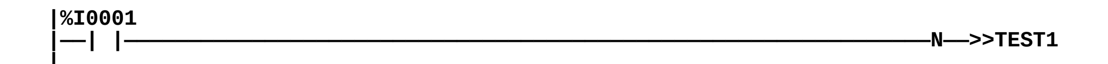
Beispiele

In den folgenden Beispielen wird der Stromfluss immer dann zu der LABEL-Funktion TEST1 weiter geschaltet, wenn die JUMP-Funktion TEST1 aktiv wird.

Beispiel einer ungeschachtelten JUMP-Funktion:



Beispiel einer geschachtelten JUMP-Funktion:



LABEL

Die LABEL-Anweisung bildet das Ziel einer JUMP-Funktion. Mit ihr wird die normale Programmbearbeitung nach einer JUMP-Funktion wieder aufgenommen.

Innerhalb eines Programms darf eine bestimmte LABEL-Bezeichnung nur einmal vorkommen. Programme ohne vollständige JUMP/LABEL-Paare können zwar erstellt und in der SPS gespeichert werden, sind aber nicht ablauffähig.

Logicmaster 90-30/20/Micro unterstützt zwei Formen der LABEL-Funktion, eine geschachtelte und eine ungeschachtelte Form. Die ungeschachtelte Form, LABEL01:, muss mit der ungeschachtelten JUMP-Funktion, ——————>>LABEL01, verwendet werden. Die geschachtelte Form, LABEL01:(geschachtelt), muss mit der geschachtelten JUMP-Funktion, ———N——>>LABEL01, verwendet werden.

Die LABEL-Funktion, die nur für sich allein in einem Strompfad stehen darf, besitzt weder Ein- noch Ausgänge.

Ungeschachtelte LABEL-Funktion:

```
???????:
```

Geschachtelte LABEL-Funktion:

```
???????: (gesch.)
```

Beispiel

In den folgenden Beispielen wird der Stromfluss von der JUMP-Funktion TEST1 bei der LABEL-Funktion TEST1 fortgesetzt.

Beispiel einer nicht geschachtelten LABEL-Funktion:

```
|
| TEST1  :
|
```

Beispiel einer geschachtelten LABEL-Funktion:

```
|
| TEST1  :(geschachtelt)
|
```

COMMENT

Mit der COMMENT-Funktion können Erläuterungen (Strompfadkommentare) in ein Programm eingefügt werden. Ein Kommentar kann dabei bis zu 2048 Zeichen umfassen. Die entsprechende Darstellung im Kontaktplanprogramm ist:



Der Text kann gelesen oder editiert werden, indem der Cursor auf (* COMMENT *) gesetzt und F10 (Zoom) gedrückt wird. Kommentartexte können auch ausgedruckt werden.

Bei längeren Kommentaren kann der Zusatztext auch in einer anderen Datei abgelegt werden. Hierzu müssen Sie folgendermaßen vorgehen:

1. Erstellen Sie den Kommentar:
   A. Geben Sie Text ein bis zu dem Punkt, an dem der Text von der anderen Datei anfangen soll.
   B. Setzen Sie den Cursor auf den Anfang einer neuen Zeile und geben dann **\I** oder **\i** ein, den Laufwerk-Kennbuchstaben gefolgt von einem Doppelpunkt, das Unterverzeichnis bzw. den Ordner und den Dateinamen. Zum Beispiel:

      \I d:\text\commnt1

      Die Laufwerkkennung kann weggelassen werden, wenn Datei und Programmordner auf dem gleichen Laufwerk liegen.
   C. Fahren Sie mit der Programmbearbeitung fort oder gehen Sie zu MS-DOS.
2. Nach Verlassen der Programmiersoftware erstellen Sie den gewünschten Text mit einer MS-DOS-kompatiblen Textverarbeitungssoftware. Weisen Sie der Textdatei den im Kommentar eingetragenen Namen zu und speichern Sie ihn auf der im Kommentar angegebenen Platte ab.

SVCREQ

Mit der SVCREQ-Funktion können Sie einen der folgenden SPS-Spezialdienste anfordern:

Tabelle 12-4. Bedienanforderungsfunktionen

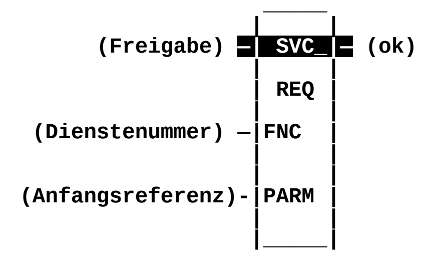
Funktion	Beschreibung
1	Zeitgeber für konstante Zyklusdauer verändern/lesen.
2	Fensterwerte lesen.
3	Zeitgeberwert und Betriebsart des Programmiergeräte-Kommunikationsfensters ändern.
4	Zeitgeberwert und Betriebsart des System-Kommunikationsfensters ändern.
6	Taskzustand und Anzahl Worte in Prüfsumme verändern/lesen
7	Echtzeituhr lesen/einstellen.
8	Überwachungszeitgeber rücksetzen.
9	Zyklusdauer seit Zyklusbeginn lesen.
10	Ordnername lesen.
11	SPS ID lesen.
12	SPS-RUN-Zustand lesen.
13	SPS abschalten.
14	Fehlertabellen löschen.
15	Letzten Fehlertabelleneintrag lesen.
16	Betriebszeituhr lesen.
18	E/A-OVERRIDE-Zustand lesen.
23	Master-Prüfsumme lesen.
26/30	E/A abfragen.
29	Verstrichene Netzausfallzeit lesen.
45	Nächsten Ausgabe- und Eingabezyklus überspringen. (Unterdrückung der E/A-Aktualisierung)
46	Status von Zugang zu schneller Rückwandplatine.

SVCREQ-Übersicht

Die SVCREQ-Funktion besitzt drei Ein- und einen Ausgangsparameter. Erhält die Funktion Stromfluss, dann wird die SPS aufgefordert, die angegebenen Funktion (FNC) auszuführen. Die Parameter für die Funktion beginnen bei der für PARM angegebenen Referenz. Die SVCREQ-Funktion gibt den Stromfluss weiter, solange keine falsche Funktionsnummer, falsche Parameter, oder außerhalb des zulässigen Bereichs liegenden Adressen angegeben wurden. Weitere Fehlerursachen werden auf den nächsten Seiten beschrieben.

Die für PARM angegebene Referenz muss in einem Wortspeicher (%R, %AI, oder %AQ) liegen. Sie ist die erste einer Gruppe, die den Parameterblock der Funktion bilden. In aufeinanderfolgenden 16-Bit-Speicherplätzen werden weitere Parameter abgelegt. Die Gesamtzahl der benötigten Referenzen hängt von dem eingesetzten SVCREQ-Funktionstyp ab.

Parameterblöcke können sowohl als Eingänge für die Funktion als auch als die Stelle, an der die Daten nach Funktionsausführung abgelegt werden, verwendet werden. Auf die von der Funktion zurückgegebenen Daten wird daher an der gleichen Adresse zugegriffen, die auch für PARM spezifiziert wurde.



Parameter

Parameter	Beschreibung
Freigabe	Die Bedienanforderung wird ausgeführt, wenn der Freigabeeingang stromführend ist.
FNC	FNC enthält die Konstante oder Referenz für den angeforderten Dienst.
PARM	PARM enthält die Anfangsadresse des Parameterblocks für den angeforderten Dienst.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde.

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
FNC		•	•	•	•		•	•	•	•	•	
PARM		•	•	•	•		•	•	•	•		
OK	•											•

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

Beispiel

Wird im folgenden Beispiel der Freigabeeingang %I0001 aktiviert, dann wird die SVCREQ-Funktion Nummer 7 mit dem bei %R0001 beginnenden Parameterblock aufgerufen. Die Ausgangsspule %Q0001 wird durchgeschaltet, wenn die Operation erfolgreich abläuft.



SVCREQ #1: Zeitgeber für konstante Zyklusdauer verändern/lesen

Ab Ausgabestand 8 der CPUs 90-30 CPU können Sie mit der SVCREQ-Funktion #1:

- Betriebsart **KONSTANTE ZYKLUSDAUER** deaktivieren.
- Betriebsart **KONSTANTE ZYKLUSDAUER** aktivieren und den alten Zeitgeberwert verwenden.
- Betriebsart **KONSTANTE ZYKLUSDAUER** aktivieren und einen neuen Zeitgeberwert verwenden.
- Nur einen neuen Zeitgeberwert einstellen.
- Den Zustand der Betriebsart **KONSTANTE ZYKLUSDAUER** und den Zeitgeberwert lesen.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen *nur* die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #1.

Der Parameterblock ist zwei Worte lang.

Um die Betriebsart **KONSTANTE ZYKLUSDAUER** zu deaktivieren, geben Sie die SVCREQ-Funktion #1 mit folgendem Parameterblock ein:

0	Adresse
wird ignoriert	Adresse + 1

Um die Betriebsart **KONSTANTE ZYKLUSDAUER** zu aktivieren, geben Sie die SVCREQ-Funktion #1 mit folgendem Parameterblock ein:

1	Adresse
0 oder Zeitgeberwert	Adresse + 1

Hinweis

Soll der Zeitgeber einen neuen Wert benutzen, geben Sie diesen in das zweite Wort ein. Soll der Zeitgeberwert nicht verändert werden, geben Sie 0 in das zweite Wort ein. Wurde zuvor noch kein Zeitgeberwert eingestellt, schaltet die Funktion den Ausgang OK ab, wenn 0 eingegeben wird.

Um den Zeitgeberwert zu verändern **ohne** die Einstellung des Zyklusmodus zu verändern, geben Sie die SVCREQ-Funktion #1 mit diesem Parameterblock ein:

2	Adresse
neuer Zeitgeberwert	Adresse + 1

Um den aktuellen Wert und Zustand des Zeitgebers zu lesen ohne dabei etwas zu verändern geben Sie die SVCREQ-Funktion #1 mit diesem Parameterblock ein:

3	Adresse
wird ignoriert	Adresse + 1

Hinweis

Nachdem die SVCREQ-Funktion #1 mit dem auf der vorherigen Seite gezeigten Parameterblock benutzt wurde, geben CPUs ab Ausgabestand 8 den Wert 0 für den Normalzyklus und den Wert 1 für konstante Zyklusdauer zurück. Verwechseln Sie dies nicht mit den nachstehend gezeigten *Eingangs*werten.

Die Funktion wird erfolgreich ausgeführt, wenn nicht

1. als angeforderte Operation eine von 0, 1, 2 oder 3 verschiedene Zahl eingegeben wird.

0	Die Betriebsart **KONSTANTE ZYKLUSDAUER** deaktivieren.
1	Die Betriebsart **KONSTANTE ZYKLUSDAUER** aktivieren.
2	Nur einen neuen Zeitgeberwert einstellen.
3	Den Zustand der Betriebsart **KONSTANTE ZYKLUSDAUER** und den Zeitgeberwert lesen. (Siehe vorstehenden Hinweis).

2. Der Zeitwert ist größer als 2550 ms (2,55 Sekunden).
3. Die konstante Zyklusdauer ist aktiviert , ohne dass ein Zeitgeberwert programmiert oder ein alter Wert von 0 für den Zeitgeber eingestellt wurde.

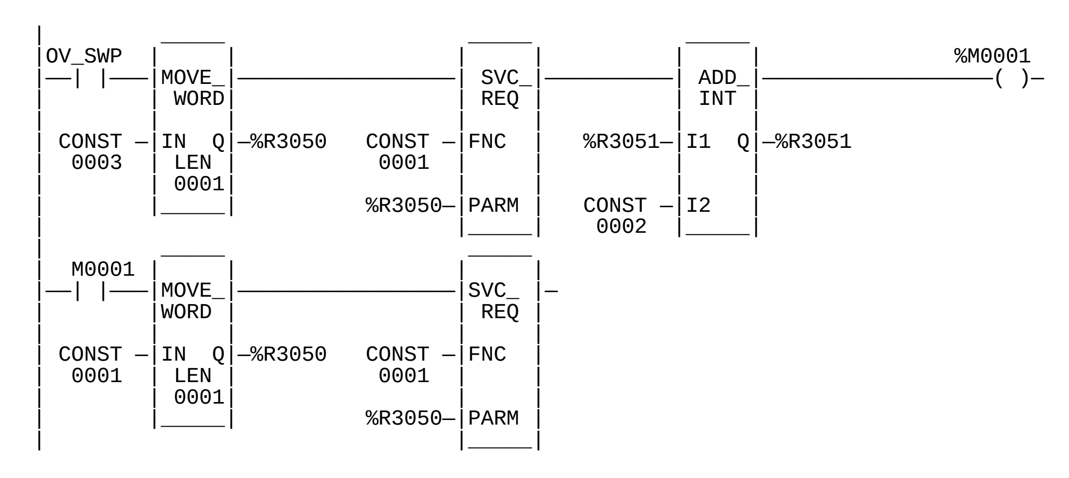
Nach der Ausführung der Funktion gibt sie den Zeitgeberstatus und -wert unter den gleichen Parameterblockadressen zurück.

0 = inaktiv	
1 = aktiv	Adresse
aktueller Zeitgeberwert	Adresse + 1

Enthält die Wortadresse + 1 den Hexadezimalwert FFFF, dann wurde bisher noch kein Zeitgeberwert programmiert.

Beispiel

Dieses Beispiel zeigt die Logik in einem Programmblock. Wird der Kontakt OV_SWP gesetzt, dann wird der Zeitgeber für konstante Zyklusdauer gelesen, um zwei Millisekunden erhöht und der neue Zeitgeberwert zur SPS zurückgeschickt. Der Parameterblock liegt im lokalen Speicher unter der Adresse %R3050. Da die Funktionen MOVE und ADD drei horizontale Kontaktpositionen benötigen, wird in dem Programmbeispiel der diskrete interne Merker %M0001 als temporäre Adresse verwendet, um das erfolgreiche Ergebnis der ersten Strompfadzeile zu halten. Bei jedem Zyklus, bei dem OV_SWP nicht gesetzt wird, wird %M0001 abgeschaltet.



SVCREQ #2: Fensterwerte lesen

Benutzen Sie die SVCREQ-Funktion #2, um die aktuellen Fenstermodus-Zeitwerte für das Programmiergeräte-Kommunikationsfenster, das System-Kommunikationsfenster und das Hintergrundtaskfenster zu erhalten.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen nur die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #2.

Für jedes Fenster gibt es drei Betriebsarten:

Modus	Wert	Beschreibung
Zeitbegrenzter Modus	0	Die Bearbeitungszeit des Fensters ist auf den jeweiligen Standardwert oder einen mit der SVCREQ-Funktion #3 (für das Programmiergeräte-Kommunikationsfenster) oder der SVCREQ-Funktion #4 (für das System-Kommunikationsfenster) festgelegten Wert begrenzt. Das Fenster ist beendet, wenn es keine Aufgaben mehr zu erfüllen hat.
Konstanter Modus	1	Jedes Fenster arbeitet in einem Modus **VOLLSTÄNDIGE BEARBEITUNG**. Während einer Zeit, die der Summe der den einzelnen Fenstern zugeordneten Zeitwerten entspricht, wechselt die CPU zwischen den drei Fenstern. Wird ein Fenster in den **KONSTANTEN MODUS** geschaltet, dann gehen die übrigen beiden Fenster automatisch ebenfalls in den **KONSTANTEN MODUS** über. Arbeitet die SPS im **KONSTANTEN FENSTERMODUS** und wurde mit der zugehörigen SVCREQ-Funktion keine spezielle Fensterbearbeitungszeit festgelegt, dann wird bei der Berechnung der konstanten Fensterzeit die Standardzeit für dieses Fenster verwendet.
Vollständige Bearbeitung	2	Unabhängig von der mit einem bestimmten Fenster verbundenen Zeit (Standardzeit oder über eine Bedienanforderungsfunktion definierte Zeit) bleibt das Fenster solange offen, bis alle Aufgaben innerhalb dieses Fensters abgeschlossen sind.

Ein Fenster wird deaktiviert, wenn der Zeitwert Null ist.

Der Parameterblock ist drei Worte lang:

	Oberes Byte	Unteres Byte	
Programmiergeräte-fenster	Modus	Wert in ms	Adresse
System-Kommunikationsfenster	Modus	Wert in ms	Adresse + 1
Hintergrundfenster			Adresse + 2

Alle Parameter sind Ausgangsparameter. Zur Programmierung dieser Funktion brauchen keine Werte in den Parameterblock eingegeben zu werden. Die Ausgangswerte für alle drei Fenster werden in Millisekunden angegeben.

Beispiel

Wird im nachstehenden Beispiel der Freigabeausgang %Q0102 gesetzt, dann trägt das SPS-Betriebssystem die aktuellen Zeitwerte der drei Fenster in den bei der Adresse %R5010 beginnenden Parameterblock ein. Weitere Beispiele mit der Funktion "Fensterwerte lesen" finden Sie in den Beschreibungen der nächsten drei SYSREQ-Funktionen.

```
|          _____
|%Q0102   |     |
|——| |———| SVC_|
|         | REQ |
|         |     |
| CONST —|FNC   |
|  0002   |     |
|         |     |
| %R5010—|PARM  |
|         |_____|
|
```

SVCREQ #3: Zeitgeberwert und Betriebsart des Programmiergeräte-Kommunikationsfensters ändern

Mit SVCREQ 3 können Sie den Modus des Programmiergeräte-Kommunikationsfensters und den Zeitgeberwert ändern. Die Änderung wird in dem CPU-Zyklus wirksam, der auf den Zyklus folgt, in dem die Funktion aufgerufen wird.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen nur die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #3.

Die SVCREQ-Funktion #3 gibt den Stromfluss nach rechts weiter, solange kein Modus 0 (zeitbegrenzt), 1 (konstante Zeit) oder 2 (vollständige Bearbeitung) eingestellt ist.

Der Parameterblock ist ein Wort lang:

Um das Programmiergerätefenster zu deaktivieren müssen Sie die SVCREQ-Funktion #3 mit diesem Parameterblock eingeben:

Oberes Byte	Unteres Byte	
0	0	Adresse

Um das Programmiergerätefenster zu aktivieren müssen Sie die SVCREQ-Funktion #3 mit diesem Parameterblock eingeben:

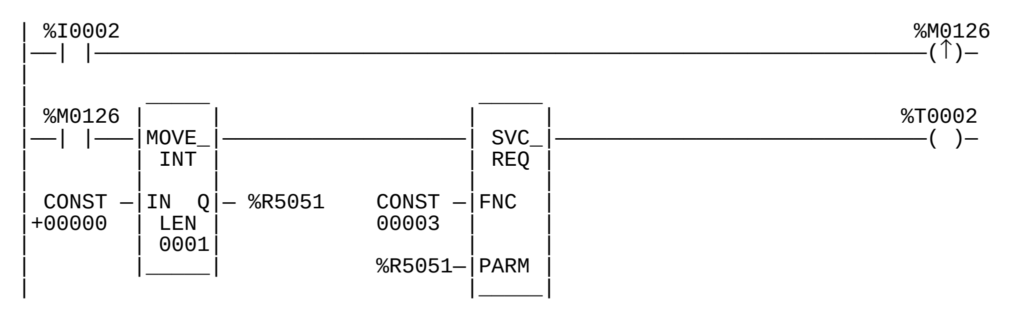
Oberes Byte	Unteres Byte	
Modus	Werte von 1 bis 255 ms	Adresse

Beispiel

Schaltet im folgenden Beispiel %M0125 auf EIN um, wird das Programmiergeräte-Kommunikationsfenster aktiviert und erhält einen Wert von 25 ms zugewiesen. Der Parameterblock steht in der Speicheradresse %R5051.

```
| %I0001                                                             %M0125
|——| |————————————————————————————————————————————————————————————————(↑)—
|
|         _____                   _____
| %M0125 |     |                 |     |                             %T0002
|——| |———|MOVE_|—————————————————| SVC_|—————————————————————————————( )—
|        | INT |                 | REQ |
|        |     |                 |     |
| CONST —|IN  Q|— %R5051  CONST —|FNC  |
|+00025  | LEN |          00003  |     |
|        | 0001|                 |     |
|        |_____|         %R5051—|PARM |
|                                |_____|
```

Weisen Sie zur Aktivierung des Programmiergeräte-Kommunikationsfensters mit der Bedienanforderungsfunktion #3 eine Wert Null (0) zu. Schaltet in diesem Beispiel %M0126 EIN, wird das Programmiergeräte-Kommunikationsfenster aktiviert und erhält einen Wert von 0 ms zugewiesen. Der Parameterblock steht in der Speicheradresse %R5051.



SVCREQ #4: Zeitgeberwert und Betriebsart des System-Kommunikationsfensters ändern

Mit SVCREQ #4 können Sie den Modus des System-Kommunikationsfensters und den Zeitgeberwert ändern. Die Änderung wird in dem CPU-Zyklus wirksam, der auf den Zyklus folgt, in dem die Funktion aufgerufen wird.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen nur die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #4.

Die SVCREQ-Funktion #4 gibt den Stromfluss nach rechts weiter, solange kein Modus 0 (zeitbegrenzt), 1 (konstante Zeit) oder 2 (vollständige Bearbeitung) eingestellt ist.

Der Parameterblock ist ein Wort lang:

Um das System-Kommunikationsfenster zu deaktivieren müssen Sie die SVCREQ-Funktion #4 mit diesem Parameterblock eingeben:

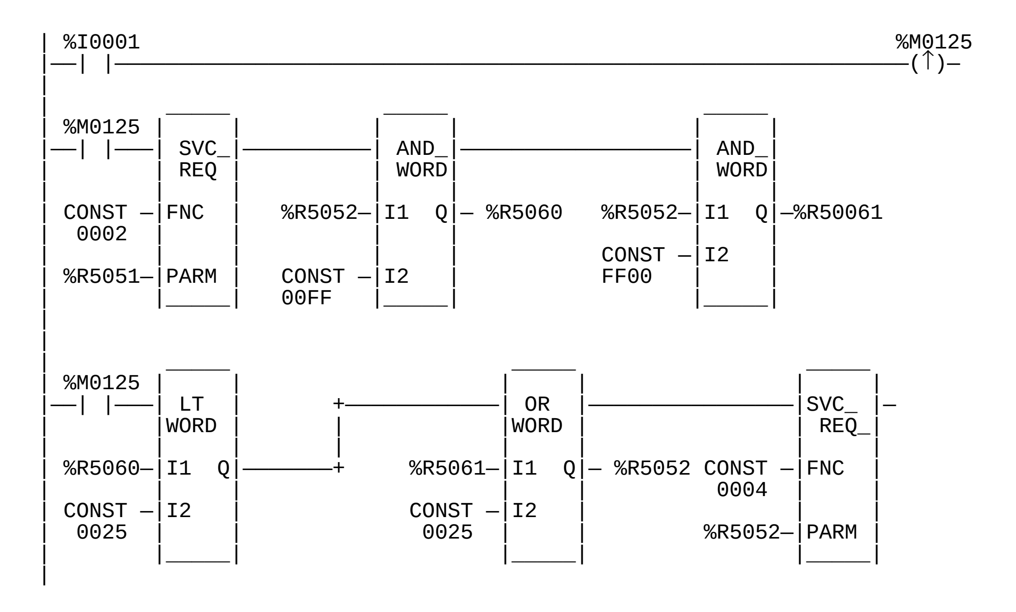
Oberes Byte	Unteres Byte	
0	0	Adresse

Um das System-Kommunikationsfenster zu aktivieren müssen Sie die SVCREQ-Funktion #4 mit diesem Parameterblock eingeben:

Oberes Byte	Unteres Byte	
Modus	Werte von 1 bis 255 ms	Adresse

Beispiel

Schaltet im folgenden Beispiel der Freigabeeingang %M0125 auf EIN um, werden Modus und Zeitgeberwert des System-Kommunikationsfensters gelesen. Der Wert wird nicht verändert, wenn der Zeitgeberwert größer als oder gleich 25 ms ist. Ein Wert, der kleiner als 25 ms ist, wird auf 25 ms gesetzt. In beiden Fällen wird das Fenster aktiviert, wenn der Strompfad die Bearbeitung beendet. Der Parameterblock für alle drei Fenster liegt bei Adresse %R5051. Da Betriebsart und Zeitgeber für das System-Kommunikationsfenster im zweiten Wert des von der Funktion "Fenswerte lesen" (Funktion #2) zurückgegebenen Parameterblocks liegen, steht die aktuelle Fensterzeit für das System-Kommunikationsfenster im unteren Byte von %R5052.



SVCREQ #6: Anzahl der Worte für Prüfsumme verändern/lesen

Mit der SVCREQ-Funktion mit der Funktionsnummer 6 können Sie:

- die aktuelle Anzahl Worte lesen;
- eine neue Wortanzahl einstellen.

Sofern bei der angeforderten Operation (siehe unten) keine von 0 oder 1 verschiedene Zahl eingegeben wird, wird die Operation erfolgreich ausgeführt.

Für die Prüfsummenfunktion hat der Parameterblock eine Länge von 2 Worten.

Aktuelle Anzahl Worte lesen:

Geben Sie die SVCREQ-Funktion #6 mit diesem Parameterblock ein:

0	Adresse
wird ignoriert	Adresse + 1

Nachdem die Funktion ausgeführt wurde, gibt sie die aktuelle Prüfsumme im zweiten Wort des Parameterblocks aus. Für die Lesefunktion wird kein Bereich angegeben, der zurückgegebene Wert gibt an, wieviel Worte derzeit durch die Prüfsummenfunktion erfasst werden.

0	Adresse
Aktuelle Anzahl Worte	Adresse + 1

Neue Wortanzahl einstellen:

Geben Sie die SVCREQ-Funktion #6 mit diesem Parameterblock ein:

1	Adresse
neue Anzahl Worte	Adresse + 1

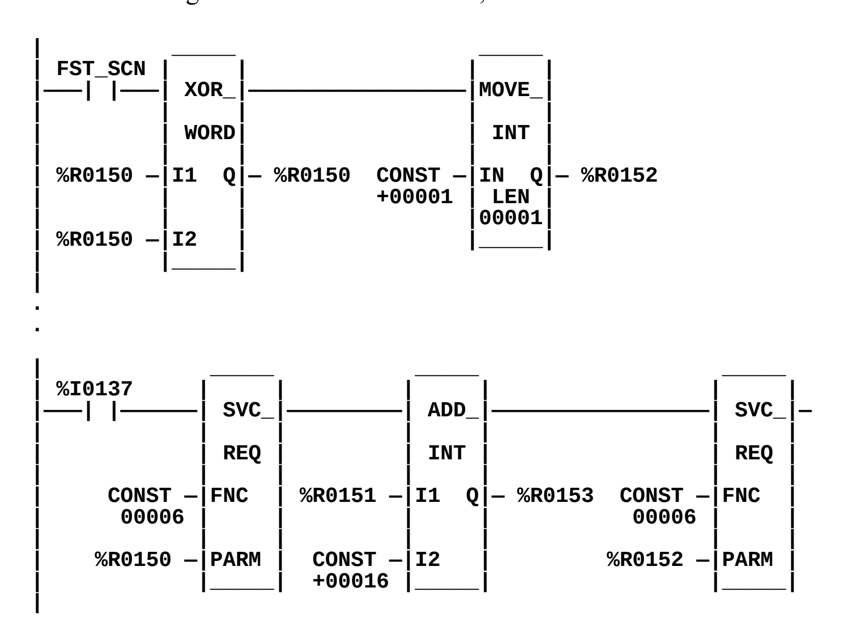
Durch Eingabe einer 1 veranlassen Sie, dass die SPS die Anzahl der durch die Prüfsummenfunktion erfassten Wörter auf den im zweiten Wort des Parameterblocks eingegebenen Wert einstellt. Bei CPU 331 oder 311 kann dieser Wert 0 oder 32 sein, bei der CPU 211 ist 0 oder 4 möglich.

Hinweis

Diese SVCREQ-Funktion ist bei der Micro-SPS nicht verfügbar.

Beispiel

Wird im nachstehenden Beispiel der Freigabekontakt FST_SCN gesetzt, werden die Parameterblöcke für die Prüfsummenfunktion aufgebaut. Später im Programm, wenn der Eingang %I0137 aktiviert wird, wird die Anzahl der in der Prüfsumme berücksichtigten Wörter aus dem SPS-Betriebssystem gelesen. Diese Anzahl wird um 16 erhöht; die Ergebnisse der Funktion ADD_INT werden dabei in den Parameter "neuen Wert für Einstellung halten" eingetragen. Der zweite Bedienanforderungsblock fordert die SPS auf, den neuen Wert einzustellen.



Die Parameterblöcke im Beispiel liegen bei Adresse %R0150 und besitzen folgenden Inhalt:

0 = aktuellen Wert lesen	%R0150
aktuellen Wert halten	%R0151

1 = aktuellen Wert einstellen	%R0152
neuen Wert für Einstellung halten	%R0153

SVCREQ #7: Echtzeituhr lesen/einstellen

Mit der SVCREQ-Funktion #7 können Sie die Echtzeituhr in der SPS lesen oder einstellen.

Hinweis

Diese Funktion ist nur verfügbar in 90-30 CPUs ab 331 und in SPS CPUs der Series 90 Micro mit 28 Punkten (d.h. IC693UDR005, IC693UAA007 und IC693UDR010) sowie den SPS CPUs der Series 90 Micro mit 23 Punkten (IC693UAL006).

Die Funktion wird erfolgreich ausgeführt, wenn nicht:

1. Eine von 0 oder 1 verschiedene Zahl als Operationskennung eingegeben wurde (siehe unten).
2. Ein unzulässiges Datenformat angegeben wurde.
3. Die angebotenen Daten nicht im erwarteten Format sind.

Bei den Datum/Zeitfunktionen ist die Länge des Parameterblocks vom Datenformat abhängig. BCD-Format benötigt 6 Worte, verdichtetes ASCII-Format 12 Worte.

0 = Zeit und Datum lesen 1 = Zeit und Datum einstellen	Adresse
1 = BCD-Format 3 = gepacktes ASCII-Format	Adresse + 1
Daten	Adresse + 2 bis Ende

In Adresse 1 wird festgelegt, ob die Funktion die Werte lesen oder verändern soll.

0 = lesen
1 = verändern

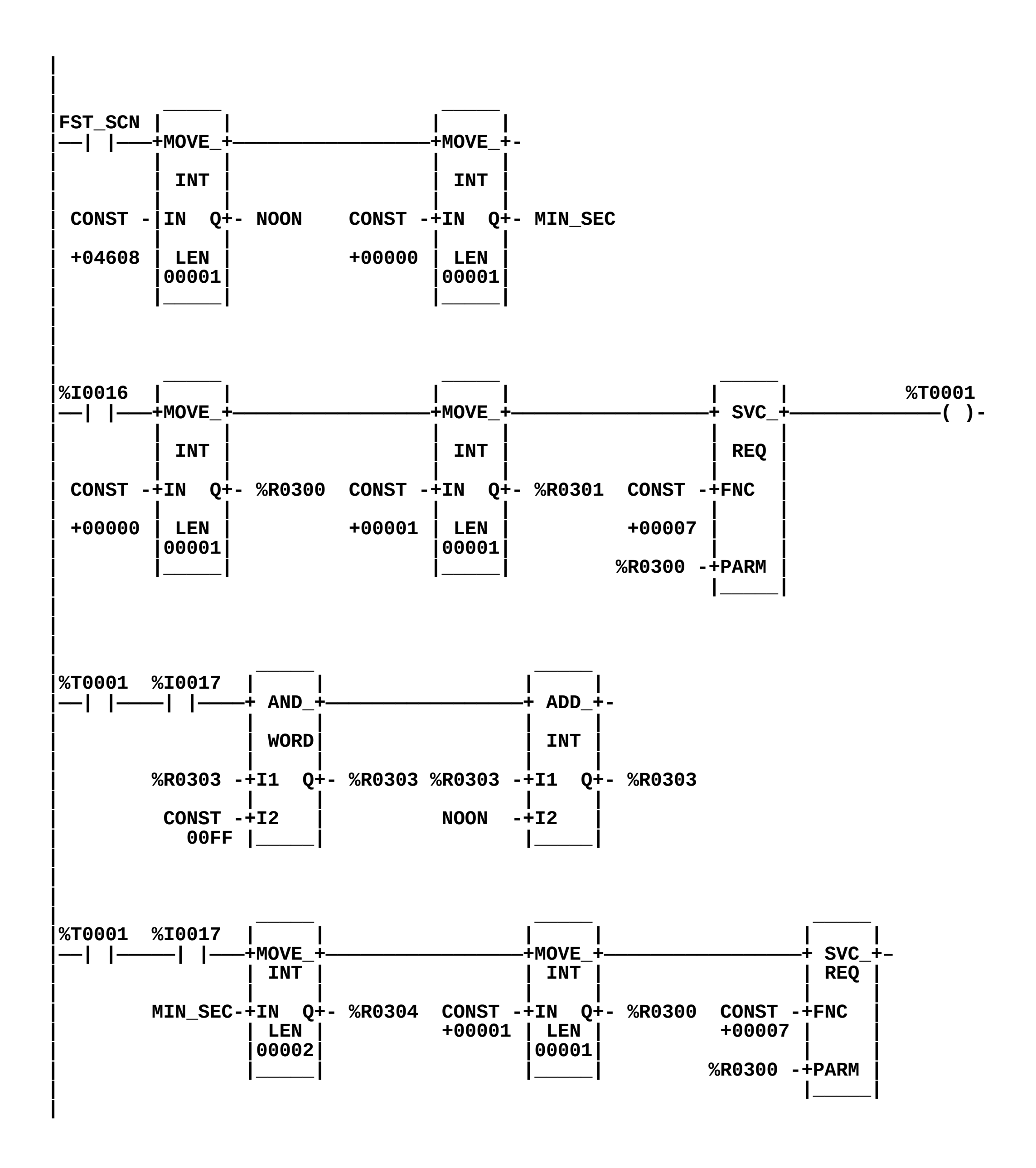
In Adresse 2 wird das Datenformat angegeben:

1 = BCD
3 = verdichtetes ASCII-Format mit eingebetteten Leerzeichen und Doppelpunkten

Die Adresse ab +2 bis zum Ende des Parameterblocks enthalten Ausgangsdaten, die von einer Lesefunktion zurückgegeben wurden, oder neue Daten, die von einer Änderungsfunktion geliefert wurden. In beiden Fällen ist das Format dieser Datenworte das gleiche. Beim Lesen von Datum und Zeit werden die Worte von (Adresse + 2) bis (Adresse +8) bei der Eingabe ignoriert.

Beispiel

Im folgenden Beispiel wird beim Aufruf von der vorhergehenden Logik ein Parameterblock für die Echtzeituhr aufgebaut, der zunächst die aktuellen Werte von Datum und Uhrzeit ausliest und dann die Uhr im BCD-Format auf 12 Uhr Mittags einstellt. Der Parameterblock liegt an der Globaladresse %R0300. Das Feld NOON wurde an beliebiger Stelle im Programm eingerichtet und enthält die Werte 12, 0 und 0 (das Feld NOON muss auch die Daten bei %R0300 enthalten). Im BCD-Format werden sechs aufeinanderfolgende Speicheradressen für den Parameterblock benötigt.



Inhalt des Parameterblocks

Auf den folgenden Seiten wird der Inhalt des Parameterblocks für die verschiedenen Datenformate dargestellt. Bei beiden Datenformaten

- werden die Stunden im 24-Stunden-Format gespeichert.
- sind die Wochentage numerische Werte:

Wert	Wochentag
1	Sonntag
2	Montag
3	Dienstag
4	Mittwoch
5	Donnerstag
6	Freitag
7	Samstag

Datum und Zeit im BCD-Format lesen/einstellen:

Im BCD-Format belegt jedes Zeit- und Datumselement ein einzelnes Byte. Dieses Format benötigt sechs Worte. Das letzte Byte des sechsten Wortes wird nicht verwendet. Beim Einstellen von Datum und Uhrzeit wird dieses Byte ignoriert. Beim Lesen von Datum und Uhrzeit gibt die Funktion 00 zurück.

Oberes Byte	Unteres Byte		Beispiel Ausgangsparameterblock: Datum und Uhrzeit im BCD-Format lesen (Sonntag, 3. Juli 1988, 14:45:30 Uhr)	
1 = ändern oder 0 = lesen		Adresse	0	
1		Adresse + 1	1	
Monat	Jahr	Adresse + 2	07	88
Stunden	Tag	Adresse + 3	14	03
Sekunden	Minuten	Adresse + 4	30	45
(Null)	Wochentag	Adresse + 5	00	01

Datum und Zeit im verdichteten ASCII-Format mit eingebetteten Doppelpunkten lesen/ändern

Im gepackten ASCII-Format ist jede Stelle der Zeit- und Datumselemente ein Byte im ASCII-Format. Darüber hinaus sind Leerzeichen und Doppelpunkte in den Daten eingebettet, um eine unveränderte Übertragung zu einem Drucker oder Sichtgerät zu ermöglichen. Dieses Format benötigt 12 Worte.

Beispiel Ausgangsparameterblock:
Datum und Zeit im gepackten ASCII-Format lesen
(Montag, 2. Oktober 1989, 23:13:00)

Oberes Byte	Unteres Byte			
1 = ändern oder 0 = lesen		Adresse	0	
3		Adresse + 1	3	
Jahr	Jahr	Adresse + 2	39	38
Monat	(Leerzeichen)	Adresse + 3	31	20
(Leerzeichen)	Monat	Adresse + 4	20	30
Tag	Tag	Adresse + 5	32	30
Stunden	(Leerzeichen)	Adresse + 6	32	20
:	Stunden	Adresse + 7	3 A	33
Minuten	Minuten	Adresse + 8	33	31
Sekunden	:	Adresse + 9	30	3 A
(Leerzeichen)	Sekunden	Adresse + 10	20	30
Wochentag	Wochentag	Adresse +11	32	30

SVCREQ #8: Zeitüberwachung rücksetzen

Mit der SVCREQ-Funktion #8 können Sie den Überwachungszeitgeber während des Zyklus rücksetzen.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen nur die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #8.

Bei Ablauf des Überwachungszeitgebers wird die SPS ohne Warnung abgeschaltet. Mit dieser Funktion kann der Zeitgeber bei einer zeitaufwendigen Aufgabe (z.B. beim Warten auf die Antwort einer Kommunikationsstrecke) am Laufen gehalten werden.

Achtung

Stellen Sie sicher, dass der Neustart des Überwachungszeitgebers keine negativen Auswirkungen auf den gesteuerten Prozess hat.

Diese Funktion besitzt zwar keinen zugehörigen Parameterblock, die Programmiersoftware fordert jedoch einen Eintrag für PARM. Sie können hier irgend eine geeignete Referenz eingeben - sie wird nicht benutzt.

Beispiel

Im folgenden Beispiel wird der Überwachungszeitgeber rückgesetzt, wenn der Freigabeausgang %Q0127, der Eingang %I1476 oder der interne Merker %M0010 gesetzt wird.

```
|                     _____
| %Q0127             |     |
|——| |———+———————————| SVC_|—
|        |           | REQ |
| %I1476 |           |     |
|——| |———|   CONST —|FNC   |
|        |    0008   |     |
| %M0010 |           |     |
|——| |———+   %AI001—|PARM  |
|                    |_____|
|
```

SVCREQ #9: Zyklusdauer seit Zyklusbeginn lesen

Mit der SVCREQ-Funktion #9 können Sie die Zeit (in Millisekunden) lesen, die seit Beginn des Zyklus verstrichen ist. Die Daten sind im 16-Bit-Wortformat.

Hinweis

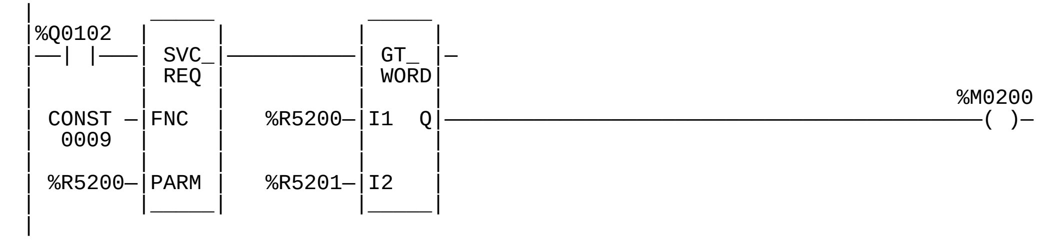
Von den in diesem Handbuch behandelten CPUs unterstützen nur die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #9.

Der Parameterblock besitzt nur Ausgangswerte. Er ist ein Wort lang.

Zeit seit Zyklusbeginn	Adresse

Beispiel

Im folgenden Beispiel wird die seit dem Anfang des Zyklus verstrichene Zeit immer in die Adresse %R0200 eingetragen. Der interne Merker %M0200 wird durchgeschaltet, wenn der Wert größer als der Wert in %R5201 ist.



SVCREQ #10: Ordnername (Programmname) lesen

Mit der SVCREQ-Funktion #10 können Sie den Namen des aktuell bearbeiteten Ordners lesen.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen *nur* die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #10.

Der Ausgangsparameterblock ist vier Worte lang. Er gibt acht ASCII-Zeichen zurück. Das letzte dieser Zeichen ist ein Nullzeichen (00h). Hat der Programmname weniger als sieben Zeichen, werden am Ende Nullzeichen hinzugefügt.

Unteres Byte	Oberes Byte	
Zeichen 1	Zeichen 2	Adresse
Zeichen 3	Zeichen 4	Adresse + 1
Zeichen 5	Zeichen 6	Adresse + 2
Zeichen 7	00	Adresse + 3

Beispiel

Wird in diesem Beispiel der Freigabeeingang %I0301 abgeschaltet, dann wird ein Wert 10 (dem Funktionscode von "Ordnername lesen") in die Registeradresse %R0099 geladen. Der Programmblock READ_ID wird dann aufgerufen, um den Ordnernamen abzufragen. Der Parameterblock liegt unter der Adresse %R0100. READ_ID wird auch im nächsten Beispiel verwendet.

```
| %I0001                                                       %I0301
|——| |——————————————————————————————————————————————————————————(↑)—
|
|          _____          __________
| %I0301 |     |         |          |
|——| |———|MOVE_|—————————| READ_ID  |
|        | WORD|         |__________|
|        |     |
| CONST —|IN  Q|— %R0099
|  0010  | LEN |
|        | 0001|
|        |_____|
|
.
.
.
Programmblock READ_ID

|
|%I0102  |     |
|——| |———| SVC_|—
|        | REQ |
|        |     |
| %R0099—|FNC  |
|        |     |
|        |     |
| %R0100—|PARM |
|        |_____|
|
```

SVCREQ #11: SPS ID lesen

Mit der SVCREQ-Funktion #11 können Sie den Namen der SPS Series 90 lesen, die das Programm bearbeitet.

Hinweis

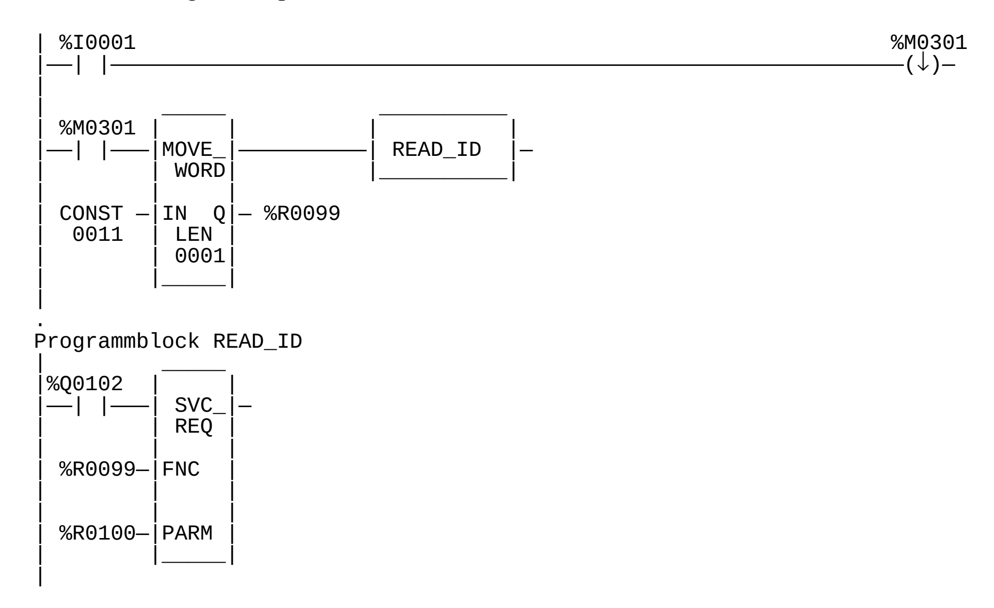
Von den in diesem Handbuch behandelten CPUs unterstützen *nur* die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #11.

Der Ausgangsparameterblock ist vier Worte lang. Er gibt acht ASCII-Zeichen zurück. Das letzte dieser Zeichen ist ein Nullzeichen (00h). Hat die SPS-Kennung weniger als sieben Zeichen, werden am Ende Nullzeichen hinzugefügt.

Unteres Byte	Oberes Byte	
Zeichen 1	Zeichen 2	Adresse
Zeichen 3	Zeichen 4	Adresse + 1
Zeichen 5	Zeichen 6	Adresse + 2
Zeichen 7	00	Adresse + 3

Beispiel

Wird in diesem Beispiel der Freigabeeingang %I0001 abgeschaltet, dann wird ein Wert 11 (dem Funktionscode von "SPS ID lesen") in die Registeradresse %R0099 geladen. Der Programmblock READ_ID wird dann aufgerufen, um die ID abzufragen. Der Parameterblock liegt unter der Adresse %R0100. Mit Ausnahme von Freigabekontakt und Funktionsnummer ist dies der gleiche Code, der im vorherigen Beispiel verwendet wurde.



SVCREQ #12: SPS-RUN-Zustand lesen

Mit der SVCREQ-Funktion #12 können Sie den aktuellen RUN-Zustand der SPS CPU auslesen.

Hinweis

Von den in diesem Handbuch behandelten CPUs unterstützen *nur* die 90-30 CPUs ab Ausgabestand 8.0 die Bedienanforderung #12.

Der Parameterblock besitzt nur Ausgangswerte und ist ein Wort lang.

1 = RUN/gesperrt	Adresse
2 = RUN/freigegeben	

Beispiel

Im folgenden Beispiel wird der RUN-Zustand der SPS immer in die Adresse %R4002 eingelesen. Ist der Zustand RUN/gesperrt, dann ruft die CALL-Funktion den Programmblock DISPLAY auf.

```
|          _____             _____
|%I0102   |     |           |     |
|——| |———| SVC_|———————————| EQ_ |—
|         | REQ |           | WORD|       __________
|         |     |           |     |      |          |
| CONST —|FNC   |   CONST —|I1  Q|——————| DISPLAY  |—
|  0012   |     |   0001    |     |      |__________|
|         |     |           |     |
| %R4002—|PARM  |   %R4002—|I2   |
|         |_____|           |_____|
|
```

SVCREQ #13: SPS abschalten (Stop)

Mit der SVCREQ-Funktion #13 wird die SPS ***am Ende des nächsten Zyklus*** abgeschaltet. Alle Ausgänge werden zu Beginn des nächsten SPS-Zyklus auf ihre vorgegebenen Zustände eingestellt. Ein Fehlerhinweis wird in der SPS-Fehlertabelle eingetragen, der darauf hinweist, dass ein Funktionsblock "SHUT DOWN PLC" [SPS abschalten] ausgeführt wurde. Der E/A-Zyklus wird entsprechend der Konfiguration fortgeführt.

Diese Funktion besitzt keinen Parameterblock.

Beispiel

Die SVCREQ-Funktion #13 wird ausgeführt, wenn beim folgenden Beispiel ein Modulverlust auftritt. Da kein Parameterblock benötigt wird, wird der PARM-Eingang nicht verwendet. Die Programmiersoftware verlangt jedoch, dass für PARM eine Angabe gemacht wird.

Dieses Beispiel benutzt eine JUMP-Funktion zum Programmende, um ein Abschalten zu erzwingen, wenn die Funktion "SPS abschalten" erfolgreich ausgeführt wird. JUMP und LABEL werden benötigt, da der Übergang in den **STOP**-Modus erst stattfindet, wenn der Zyklus, in dem die Funktion ausgeführt wurde, beendet ist.

```
|
|LOS_MD                                                        %T0001
|
|—| |——————————————————————————————————————————————————————————(↑)—
|
|              _____
|%T0001       |     |
|—| |—————————| SVC_|——————————————————————————————————————>> END_PRG
|             |     |
|             | REQ |
|             |     |
|     CONST  —|FNC  |
|       0013  |     |
|             |     |
|    %R1001  —|PARM |
|             |_____|
|
.
.
.
|
| END_PRG:
|
|
| [      END OF PROGRAM LOGIC     ]
|
```

Hinweis

Um sicherzustellen, dass der Kontakt %S0002 LST_SCN ordnungsgemäß arbeitet, führt die SPS nach dem Zyklus, in dem die SVCREQ-Funktion #13 ausgeführt wurde, einen zusätzlichen Zyklus aus.

SVCREQ #14: Fehlertabellen löschen

Mit der SVCREQ-Funktion #14 können SPS- oder E/A-Fehlertabelle gelöscht werden. Der SVCREQ-Ausgang wird durchgeschaltet, solange als geforderte Operation keine von 0 oder 1 verschiedene Zahl eingegeben wird (siehe unten).

Bei dieser Funktion ist der Parameterblock ein Wort lang. Der Parameterblock hat nur Eingänge.

0 = SPS-Fehlertabelle löschen. 1 = E/A-Fehlertabelle löschen.	Adresse

Beispiel

In diesem Beispiel wird die SPS-Fehlertabelle gelöscht, wenn die Eingänge %I0346 und %I0349 beide stromführend sind. Die E/A-Fehlertabelle wird gelöscht, wenn die Eingänge %I0347 und %I0349 beide stromführend sind. Beide Fehlertabellen werden gelöscht, wenn die Eingänge %I0348 und %I0349 stromführend sind.

Der Parameterblock für die SPS-Fehlertabelle steht in %R0500, der für die E/A-Fehlertabelle in %R0550. Beide Parameterblöcke werden an anderer Stelle im Programm eingerichtet.

```
|                         _____
|%I0349  %I0346          |     |
|--| |--+--| |--+--------| SVC_|-
|       |       |        |     |
|       |       |        | REQ |
|       |%I0348 |        |     |
|       +--| |--+CONST --|FNC  |
|                 0014   |     |
|                        |     |
|               %R0500 --|PARM |
|                        |_____|
|
|                         _____
|%I0349  %I0347          |     |
|--| |--+--| |--+--------| SVC_|-
|       |       |        |     |
|       |       |        | REQ |
|       |%I0348 |        |     |
|       +--| |--+CONST  -|FNC  |
|                 0014   |     |
|                        |     |
|               %R0550  -|PARM |
|                        |_____|
|
```

SVCREQ #15: Letzten Fehlertabelleneintrag lesen

Mit der SVCREQ-Funktion #15 können Sie den zuletzt gemachten Eintrag in die SPS- oder E/A-Fehlertabelle lesen. Der SVCREQ-Ausgang wird durchgeschaltet, sofern nicht eine von 0 oder 1 verschiedene Zahl als gewünschte Operation eingegeben wird (siehe unten) oder die Fehlertabelle leer ist. Weitere Informationen über Fehlertabelleneinträge finden Sie in Kapitel 3 "Fehlerbeschreibungen und Fehlerbehebung.

Bei dieser Funktion ist der Parameterblock 22 Worte lang. Der Eingangsparameterblock besitzt folgendes Format:

0 = SPS-Fehlertabelle lesen. 1 = E/A-Fehlertabelle lesen.	Adresse

Das Format des Ausgangsparameterblocks hängt davon ab, ob die Funktion die Daten aus der SPS-Fehlertabelle oder der E/A-Fehlertabelle liest.

Ausgabeformat für SPS-Fehlertabelle: Unteres Byte	Oberes Byte		Ausgabeformat für E/A-Fehlertabelle: Unteres Byte	Oberes Byte
0			1	
lang/kurz		Adresse + 1	lang/kurz	
Reserve		Adresse + 2	Referenzadresse	
SPS-Fehleradresse		Adresse + 3	E/A-Fehleradresse	
		Adresse + 4		
Fehlergruppe und Aktion		Adresse + 5		
Fehlercode		Adresse + 6	Fehlergruppe und Aktion	
		Adresse + 7	Fehlerkategorie	Fehlertyp
		Adresse + 8	Fehlerbeschreibung	
		Adresse + 9	fehlerspezifische Daten	
		Adresse + 10		
		Adresse +11		
fehlerspezifische Daten		Adresse + 12		
		Adresse + 13		
		Adresse + 14		
		Adresse + 15		
		Adresse + 16		
		Adresse + 17		
		Adresse + 18		
Zeitstempel		Adresse + 19	Zeitstempel	
		Adresse + 20		
		Adresse + 21		

Im ersten Byte von Wortadresse + 1 gibt "kurz/lang" die Anzahl der fehlerspezifischen Daten im Fehlereintrag an. Es bedeuten:

SPS-Fehlertabelle	**00 = -8 Byte (kurz)**
	01 = 24 Byte (lang)
E/A-Fehlertabelle:	**02 = –5 Byte (kurz)**
	03 = 21 Byte (lang)

Beispiel 1

Sind im folgenden Beispiel die Eingänge %I0251 und %I0250 aktiviert, wird der letzte Eintrag der SPS-Fehlertabelle in den Parameterblock eingelesen. Ist Eingang %I0251 AUS und Eingang %I0250 EIN, dann wird der letzte Eintrag der E/A-Fehlertabelle in den Parameterblock eingelesen. Der Parameterblock liegt unter der Adresse %R0600.

```
|
|                   _____
|%I0250  %I0251   |     |
|——| |—————| |———|MOVE_|
|                 |     |
|                 | INT |
|                 |     |
|         CONST —|IN  Q|— %R0600
|           0000  | LEN |
|                 | 0001|
|                 |_____|
|
|                   _____
|%I0250  %I0251   |     |
|——| |—————|/|———|MOVE_|
|                 |     |
|                 | INT |
|                 |     |
|         CONST —|IN  Q|— %R0600
|           0001  | LEN |
|                 | 0001|
|                 |_____|
|
|          _____
|ALW_ON   |     |
|——| |———| SVC_|—
|         |     |
|         | REQ |
|         |     |
| CONST —|FNC  |
|   0015  |     |
|         |     |
|%R0600 —|PARM |
|         |_____|
|
```

Beispiel 2

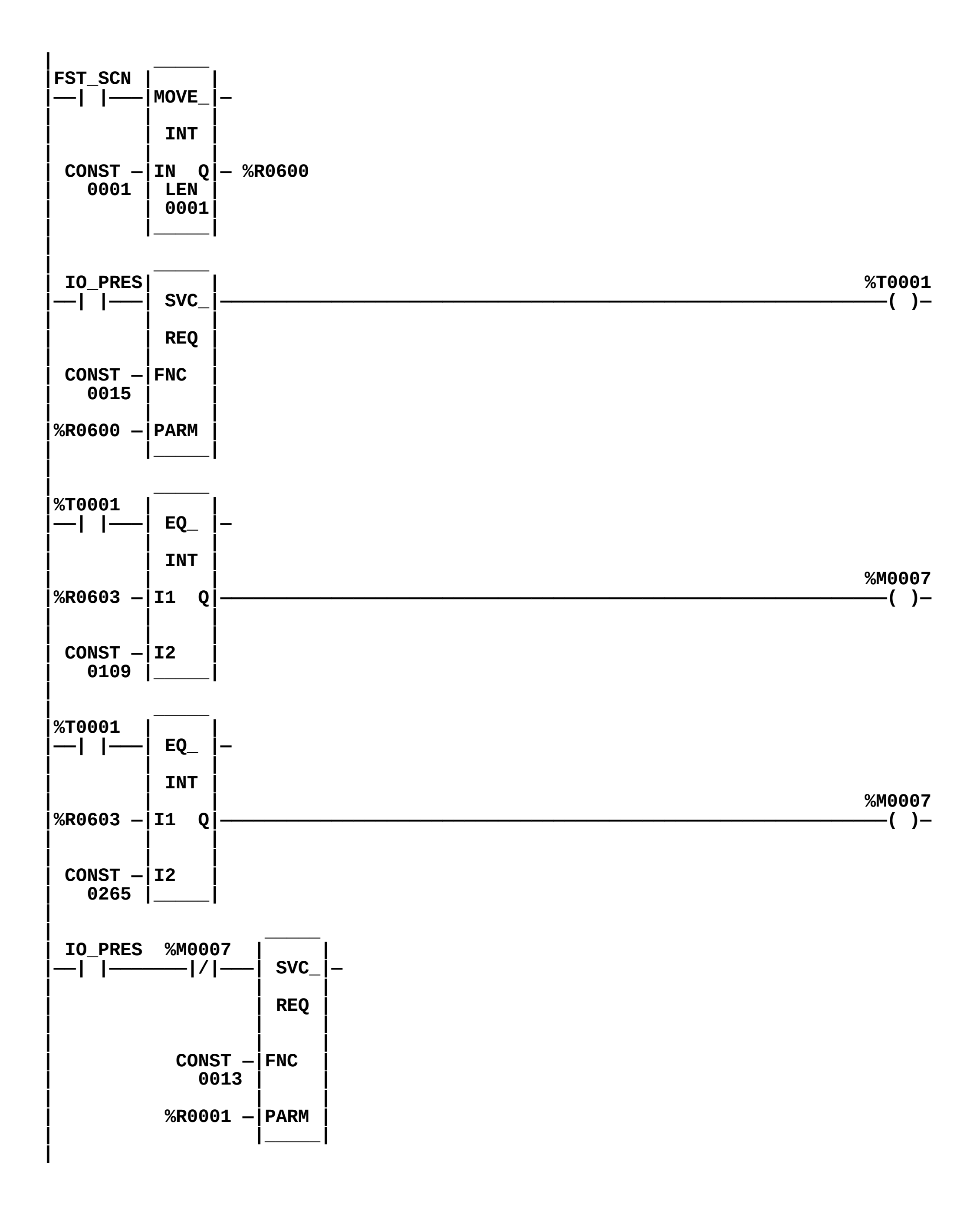
Im folgenden Beispiel wird die SPS abgeschaltet, wenn ein Fehler auf einem E/A-Modul auftritt (Ausnahme: Fehler auf Modulen in Chassis 0, Steckplatz 9 oder Chassis 1, Steckplatz 9). Tritt ein Fehler in einem dieser beiden Module auf, dann läuft das System weiter. Der Parameter für "Tabellentyp" wird im ersten Zyklus eingestellt. Ist der Kontakt IO-PRES gesetzt, enthält die E/A-Fehlertabelle einen Eintrag. Die SPS-CPU setzt den Schließerkontakt in dem auf den Fehlereintrag folgenden Zyklus. Werden bei zwei aufeinanderfolgenden Zyklen Fehler eingetragen, wird der Schließerkontakt bei zwei aufeinanderfolgenden Zyklen gesetzt.

Das Beispiel verwendet einen Parameterblock in %R0600. Nachdem die SVCREQ-Funktion ausgeführt wurde, steht die Moduladresse im 4., 5. und 6. Wort des Parameterblocks:

1		%R0600
lang/kurz		%R0601
Referenzadresse		%R0602
Chassisnummer	Steckplatznummer	%R0603
E/A-Bus-Nr.	Busadresse	%R0604
Punktadresse		%R0605

Fehlerdaten

Im Programm vergleichen die EQ-INT-Blöcke die Chassis-/Steckplatzadresse in der Tabelle mit den hexadezimalen Konstanten. Der interne Merker %M00007 wird durchgeschaltet, wenn Chassis und Steckplatz mit dem fehlerhaften Modul die vorgenannten Kriterien erfüllen. Ist %M00007 durchgeschaltet, dann ist sein Öffnerkontakt AUS und verhindert die Abschaltung. Ist dagegen %M00007 AUS, da der Fehler auf einem anderen Modul aufgetreten ist, dann ist der Öffnerkontakt EIN und die SPS wird abgeschaltet.


FST_SCN
MOVE_
INT
CONST
0001
IN Q
LEN
0001
%R0600
IO_PRES
SVC_
REQ
CONST
0015
FNC
%R0600
PARM
%T0001
%T0001
EQ_
INT
%R0603
I1 Q
CONST
0109
I2
%M0007
%T0001
EQ_
INT
%R0603
I1 Q
CONST
0265
I2
%M0007
IO_PRES
%M0007
SVC_
REQ
CONST
0013
FNC
%R0001
PARM

SVCREQ #16: Betriebszeituhr lesen

Mit der SVCREQ-Funktion #16 kann der Wert der Betriebszeituhr ausgelesen werden. Beginnend mit dem Zeitpunkt, zu dem die SPS eingeschaltet wird, erfasst diese Uhr die Zeit in Sekunden. Der Erfassungszeitraum dieser Uhr beträgt ungefähr 100 Jahre.

Diese Funktion besitzt nur einen Ausgangsparameterblock Der Parameterblock ist drei Worte lang.

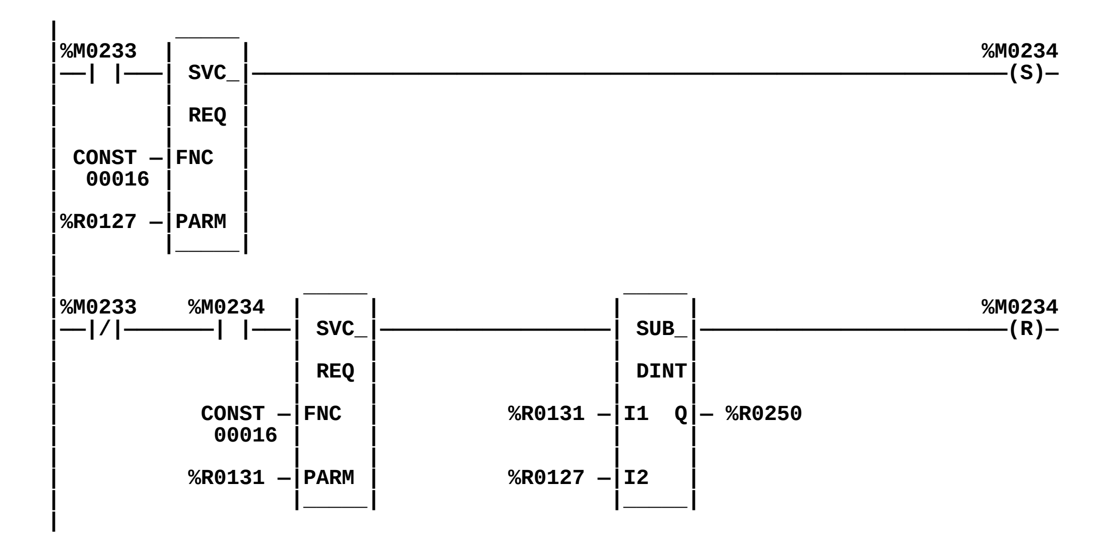
Sekunden seit Einschalten (niedrigstwertig)	Adresse
Sekunden seit Einschalten (höchstwertig)	Adresse + 1
100-Mikrosekunden-Zeitimpulse	Adresse + 2

Die ersten beiden Worte geben die verstrichene Zeit in Sekunden an. Das letzte Wort ist die Anzahl der 100-Mikrosekunden-Zeitimpulse der aktuellen Sekunde.

Beispiel

Wird beim folgenden Beispiel der interne Merker %M0233 durchgeschaltet, dann wird der Wert der Betriebszeituhr gelesen und die Spule %M0234 gesetzt. Der Wert wird erneut gelesen, wenn der interne Merker abfällt. Der Unterschied zwischen den beiden Werten wird dann berechnet und das Ergebnis im Registerspeicher unter der Adresse %R0250 abgelegt.

Der Parameterblock beim ersten Lesen ist %R0127, beim zweiten Lesen %R0131. Bei der Berechnung werden die 100-Mikrosekunden-Anteile und die Tatsache, dass der DINT-Type eigentlich ein vorzeichenbehafteter Wert ist, vernachlässigt. Die Berechnung ist über einen Zeitraum von etwa 50 Jahren korrekt.



SVCREQ #18: E/A-OVERRIDE-Zustand lesen

Mit der SVCREQ-Funktion #18 kann der aktuelle Override-Zustand in der CPU gelesen werden.

Hinweis

Diese Funktion ist ***nur*** ab CPU 331 aufwärts verfügbar.

Bei dieser Funktion ist der Parameterblock ein Wort lang. Der Parameterblock hat nur Ausgänge.

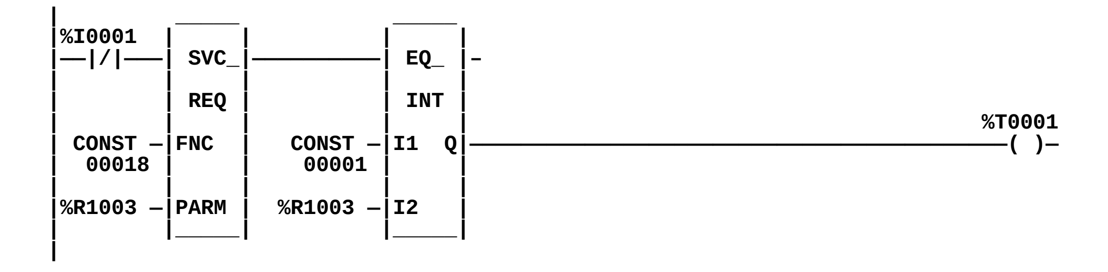
0 = Kein OVERRIDE-Zustand eingestellt. 1 = OVERRIDE-Zustände eingestellt.	Adresse

Hinweis

SVCREQ #18 meldet nur Override-Zustände der %I- und %Q-Referenzen.

Beispiel

Im folgenden Beispiel wird der Zustand der E/A-Overrides immer in Adresse %R1003 eingelesen. Ausgang %T0001 wird durchgeschaltet, wenn Override-Zustände bestehen.



SVCREQ #23: Master-Prüfsumme lesen

Mit der SVCREQ-Funktion #23 können Sie die Master-Prüfsummen von Anwenderprogramm und Konfiguration lesen. Solange die Funktion freigegeben ist, ist der SVCREQ-Ausgang immer EIN; der Ausgangsdatenblock (siehe unten) beginnt an der in Parameter 3 (PARM) der SVCREQ-Funktion angegebenen Adresse.

Während eines **SPEICHERN IM RUN-MODUS** können die Programm-Prüfsummen bis zum Ende des Speicherns ungültig sein. Zwei Merker am Anfang des Ausgangsparameterblocks zeigen daher an, ob Programm- und Konfigurations-Prüfsummen gültig sind.

Der Ausgangsparameterblock für diese Funktion ist 12 Worte lang und hat folgendes Format:

Haupt-Programmprüfsumme gültig (0 = nicht gültig, 1 = gültig)	Adresse
Haupt-Konfigurationsprüfsumme gültig (0 = nicht gültig, 1 = gültig)	Adresse + 1
Anzahl Programmblöcke (including _MAIN)	Adresse + 2
Anwenderprogrammgröße in Byte (Datentyp DWORD)	Adresse + 3
Additive Prüfsumme von Programm	Adresse + 5
CRC-Prüfsumme von Programm (Datentyp DWORD)	Adresse + 6
Länge der Konfigurationsdaten in Byte	Adresse + 8
Additive Prüfsumme der Konfiguration	Adresse + 9
CRC-Prüfsumme von Konfiguration (Datentyp DWORD)	Adresse + 10

Beispiel

Wird im nachstehenden Beispiel Eingang %I0251 aktiviert, dann werden die Daten zur Master-Prüfsumme in den Parameterblock eingetragen und die Ausgangsspule %Q0001 wird durchgeschaltet. Der Parameterblock liegt bei %R0050.

```
|          _____
|%I0251   |     |                                                   %Q0001
|--| |----| SVC_|---------------------------------------------------( )-
|         |     |
|         | REQ |
|         |     |
| CONST  -|FNC  |
|  +0023  |     |
|         |     |
|%R0050  -|PARM |
|         |_____|
|
```

SVCREQ #26/30: E/A abfragen

Mit SVCREQ-Funktion #26 (oder #30 — beide Funktionen sind identisch und liefern das gleiche Ergebnis) können Sie die vorhandenen Module abfragen und mit der Chassis/Steckplatz-Konfiguration vergleichen und die gleichen Alarme für hinzugefügte, verlorengegangen oder falsch konfigurierte Module erzeugen, wie sie bei einer Konfigurationsspeicherung hervorgerufen werden. Je nach Fehler erzeugt diese SVCREQ-Funktion Fehlermeldungen für die SPS- oder die E/A-Fehlertabelle.

Diese Funktion besitzt keinen Parameterblock und liefert immer Stromfluss am Ausgang.

Hinweis

Die von dieser SVCREQ-Funktion benötigte Ausführungszeit hängt von der Anzahl der vorliegenden Fehler ab und steigt mit der Anzahl fehlerhafter Module.

Beispiel

Wird im nachstehenden Beispiel Eingang %I0251 aktiviert, werden die vorhandenen Module abgefragt und mit der Chassis-/Steckplatz-Konfiguration verglichen. Ausgang %Q0001 wird durchgeschaltet, wenn die SVCREQ beendet ist.



Hinweis

Diese SVCREQ-Funktion ist bei der Micro-SPS nicht verfügbar.

SVCREQ #29: Verstrichene Netzausfallzeit lesen

Mit der SVCREQ-Funktion #29 können Sie die Zeit auslesen, die zwischen dem letzten Abschalten und dem letzten Einschalten der Versorgungsspannung verstrichen ist. Der SVCREQ-Ausgang ist immer durchgeschaltet und der Ausgangsdatenblock (siehe unten) beginnt an der in Parameter 3 (PARM) der SVCREQ-Funktion eingestellten Adresse.

Hinweis

Diese Funktion steht nur in CPUs ab Modell 331 zur Verfügung.

Diese Funktion besitzt nur einen Ausgangsparameterblock Der Parameterblock ist drei Worte lang.

Netzausfallzeit in Sekunden (niedrigstwertig)	Adresse
Netzausfallzeit in Sekunden (höchstwertig)	Adresse + 1
100-Mikrosekunden-Zeitimpulse	Adresse + 2

Die ersten beiden Worte geben die Netzausfallzeit in Sekunden an. Das letzte Wort gibt die restliche Netzausfallzeit in 100-Mikrosekunden-Anteilen an (ist immer Null). Kann die SPS die aufgelaufene Netzausfallzeit nicht richtig berechnen, wird die Zeit auf 0 gesetzt. Dies ist der Fall, wenn beim Einschalten der SPS am HHP CLR M/T gedrückt wird oder wenn die Zeitüberwachung (Watchdog) vor dem Abschalten anspricht.

Beispiel

Wird im folgenden Beispiel Eingang %I0251 aktiviert, dann wird die aufgelaufene Netzausfallzeit in den Parameterblock eingetragen und die Ausgangsspule %Q0001 wird durchgeschaltet. Der Parameterblock liegt bei %R0050.

```
|           _____
|%I0251    |     |                                          %Q0001
|--| |-----| SVC_|--------------------------------------------( )-
|          |     |
|          | REQ |
|          |     |
| CONST  - |FNC  |
|  +0029   |     |
|          |     |
|%R0050  - |PARM |
|          |_____|
|
```

SVCREQ #45: Nächsten Ausgabe- und Eingabezyklus überspringen

(Unterdrückung der E/A-Aktualisierung) Mit der SVCREQ-Funktion #45 können Sie die nächsten Ausgabe- und Eingabezyklen überspringen. Alle Änderungen der Ausgangsreferenztabellen, die während des Zyklus auftreten, in dem die SVCREQ-Funktion #45 ausgeführt wird, wirken sich nicht auf die physikalischen Ausgänge der entsprechenden Module aus. Alle Änderungen der physikalischen Eingangsdaten der Module haben während des Zyklus, der auf den Zyklus folgt, in dem die SVCREQ-Funktion #45 ausgeführt wird, keine Auswirkung auf die entsprechenden Eingangsreferenzen.

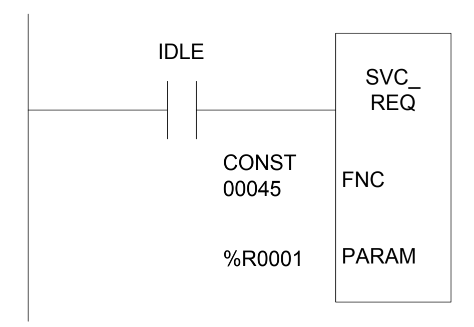
Diese Funktion besitzt keinen Parameterblock.

Hinweis

Der DOIO-Funktionsblock ist vom Einsatz der SVCREQ-Funktion #45 nicht betroffen. Er führt die Aktualisierung der E/A fort, selbst wenn er im gleichen Logikprogramm wie die SVCREQ-Funktion #45 verwendet wird.

Beispiel

Im folgenden Beispiel wird der nächste Ausgangs- und Eingangszyklus übersprungen, wenn der Kontakt "Idle” Stromfluss weiter schaltet.



SVCREQ #46: Schneller Rückwandplatinen-Statuszugriff

Mit der SVCREQ-Funktion #46 können Sie eine der folgenden schnellen Rückwandplatinen-Zugriffsfunktionen ausführen:

1. Ein Wort Statuszusatzdaten von einem oder mehreren spezifizierten intelligenten Modulen lesen.
2. Ein Wort Statuszusatzdaten zu einem oder mehreren spezifizierten intelligenten Modulen schreiben.
3. Lesen/schreiben: In einer einzigen Operation ein Wort Statuszusatzdaten von einem oder mehreren spezifizierten Modulen lesen und einen Datenwert zwischen 0 und 15 zum gleichen Modul schreiben.

Bemerkung

Diese Bedienanforderung kann nur bei Modulen angewandt werden, die sie unterstützen. Derzeit wird die Funktion nur von einem einzigen Modul, dem DSM (digitales Servomodul) Version 312 unterstützt, das zum Zeitpunkt der Veröffentlichung dieses Handbuchs noch nicht lieferbar ist. Die Freigabe soll jedoch in Kürze erfolgen.

In dem Programmzyklus, in dem eine der Schreibfunktionen ausgeführt wird, sollte mit den angegebenen Modulen nicht gleichzeitig ein COMM_REQ- oder DOIO-Funktionsblock ausgeführt werden. Hierdurch können die zu schreibenden Daten verloren gehen.

Zwei Funktionen, die zu einem Modul schreiben (schreiben oder lesen/schreiben) sollten nicht im selben Programmzyklus auf das selbe Modul angewandt werden. Hierdurch können die ersten Schreibdaten verloren gehen.

Diese Bedienanforderung besitzt eine variable Länge (siehe unten). Das erste Wort des Parameterblocks legt fest, welche Funktion verwendet wird. Es besitzt das folgende Format:

1 = Zusatzdaten lesen 2= Zusatzdaten schreiben 3 = Zusatzdaten lesen/schreiben	Adresse

Statuszusatzdaten lesen (Funktion #1)

Die Funktion "Zusatzdaten lesen" liest ein Wort Statuszusatzdaten von jedem der in einer Liste im Parameterblock spezifizierten Module und legt die Statusdatenwerte im Parameterblock ab. Der Parameterblock belegt (N + 4) Worte im Referenzspeicher. N ist die Anzahl Module, zu denen die Daten geschrieben werden.

Benutzen Sie die Tabelle auf der folgenden Seite zur Interpretation der Ausgangswerte.

Tabelle 12-5. Ausgangswerte für die Funktion "Zusatzdaten lesen"

Adresse	Feld	Bedeutung
Adresse	Funktion	1 = Statuszusatzdaten lesen
Adresse + 1	Fehlercode	Hier wird ein Fehlercode eingetragen, wenn die Funktion wegen eines fehlenden, unpassenden oder defekten Moduls fehlschlägt. Einzelheiten siehe "Fehlercodes" auf Seite 12-69.
Adresse + 2	Chassis und Steckplatz von Fehler	Gibt an, bei welcher Chassis- und Steckplatznummer der Fehler aufgetreten ist.
Adresse + 3	1. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des ersten Moduls, aus dem die Daten gelesen werden.
Adresse + 4	Daten vom 1. Modul lesen	Hier werden die aus dem ersten Modul gelesenen Daten abgelegt.
Adresse + 5	2. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des zweiten Moduls, aus dem die Daten gelesen werden.
Adresse + 6	Daten vom 2. Modul lesen	Hier werden die aus dem zweiten Modul gelesenen Daten abgelegt.
Adresse + (I * 2) + 1	I. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des I-ten Moduls, aus dem die Daten gelesen werden.
Adresse + (I * 2) + 2	Daten vom I-ten Modul lesen	Hier werden die aus dem I-ten Modul gelesenen Daten abgelegt.
Adresse + (N * 2) + 1	Letztes Chassis und letzter Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des letzten Moduls, aus dem die Daten gelesen werden.
Adresse + (N * 2) + 2	Daten vom letzten Modul lesen	Hier werden die aus dem letzten Modul gelesenen Daten abgelegt.
Adresse + (N * 2) + 3	Ende der Listenanzeige.	Eine Null in diesem Wort zeigt das Ende der Modulliste an.

Daten schreiben (Funktion #2)

Die Funktion "Daten schreiben" schreibt einen Datenwert zwischen 0 und 15 aus dem Parameterblock in Module, die über eine Liste im Parameterblock festgelegt werden. Der Parameterblock belegt (N + 4) Worte im Referenzspeicher. N ist die Anzahl Module, zu denen die Daten geschrieben werden.

Tabelle 12-6. Ausgangswerte für die Funktion "Daten schreiben"

Adresse	Feld	Bedeutung
Adresse	Funktion	2= Daten schreiben
Adresse + 1	Fehlercode	Hier wird ein Fehlercode eingetragen, wenn die Funktion wegen eines fehlenden, unpassenden oder defekten Moduls fehlschlägt. Es wird kein Fehlercode eingetragen, wenn die Funktion zwar ausgeführt wird, aber die Module die geschriebenen Daten nicht fehlerfrei empfangen. Einzelheiten siehe "Fehlercodes" auf Seite 12-69.
Adresse + 2	Chassis und Steckplatz von Fehler	Gibt an, bei welcher Chassis- und Steckplatznummer der Fehler aufgetreten ist.
Adresse + 3	1. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des ersten Moduls, zu dem die Daten geschickt werden.
Adresse + 4	Daten für erstes Modul schreiben.	Dieser Datenwert wird zum ersten Modul geschrieben.
Adresse + 5	2. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des zweiten Moduls, zu dem die Daten geschickt werden.
Adresse + 6	Daten für zweites Modul schreiben.	Dieser Datenwert wird zum zweiten Modul geschrieben.
Adresse + (I * 2) + 1	I. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des I-ten Moduls, zu dem die Daten geschickt werden.
Adresse + (I * 2) + 2	Daten für I-tes Modul schreiben.	Dieser Datenwert wird zum I-ten Modul geschrieben.
Adresse + (N * 2) + 1	Letztes Chassis und letzter Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des letzten Moduls, zu dem die Daten geschickt werden.
Adresse + (N * 2) + 2	Daten für letztes Modul schreiben.	Dieser Datenwert wird zum letzten Modul geschrieben.
Adresse + (N * 2) + 3	Ende der Listenanzeige.	Eine Null in diesem Wort zeigt das Ende der Modulliste an.

Daten lesen/schreiben (Funktion #3)

Die Funktion "Daten lesen/schreiben" liest ein Wort Statuszusatzdaten aus einem im Parameterblock angegebenen Modul und schreibt dann in dieses Modul einen Datenwert zwischen 0 und 15 aus dem Parameterblock. Dieser Lese-/Schreibvorgang wird für jedes Modul in einer Liste im Parameterblock wiederholt. Der Parameterblock belegt (N * 3) + 3 Worte im Referenzspeicher. N ist die Anzahl Module, mit denen Daten ausgetauscht werden.

Tabelle 12-7. Ausgangswerte für die Funktion "Daten lesen/schreiben"

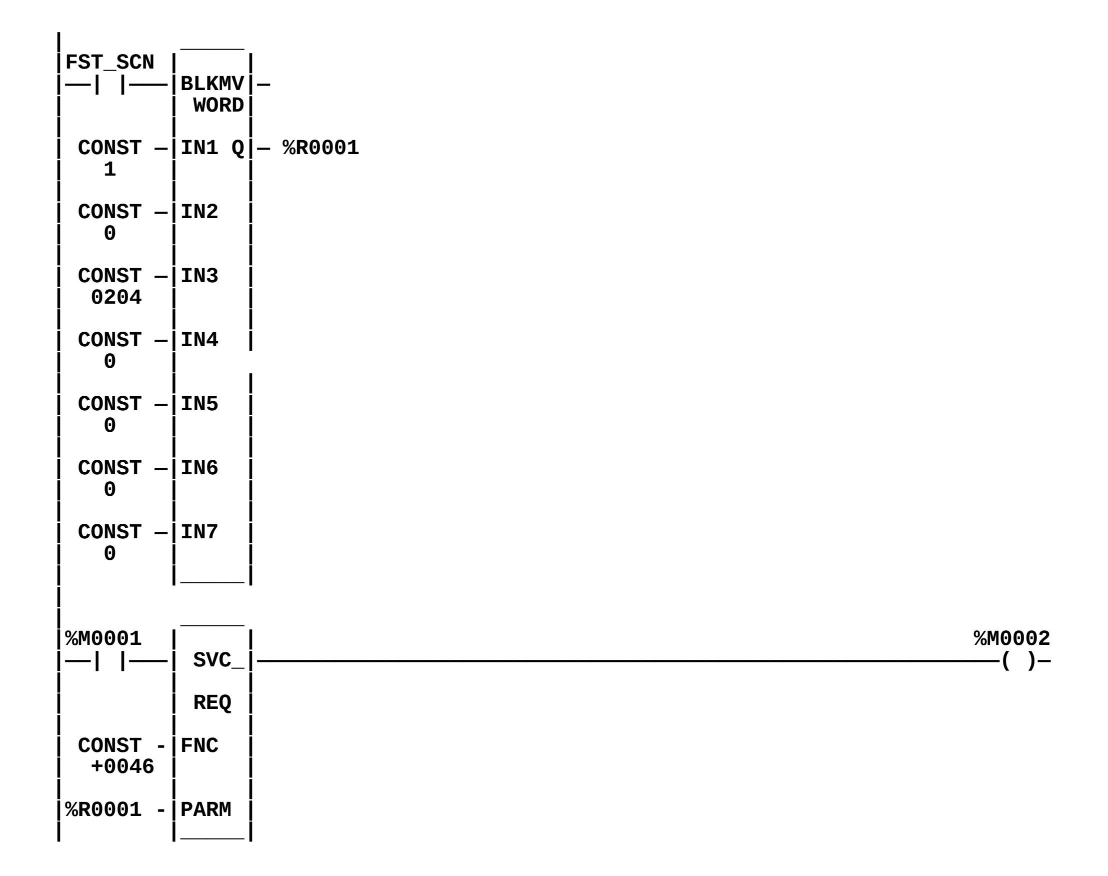
Adresse	Feld	Bedeutung
Adresse	Funktion	3 = lesen/schreiben
Adresse + 1	Fehlercode	Hier wird ein Fehlercode eingetragen, wenn die Funktion wegen eines fehlenden, unpassenden oder defekten Moduls fehlschlägt. Es wird kein Fehlercode eingetragen, wenn die Funktion zwar ausgeführt wird, aber die Module die geschriebenen Daten nicht fehlerfrei empfangen. Einzelheiten siehe "Fehlercodes" auf Seite 12-69.
Adresse + 2	Chassis und Steckplatz von Fehler	Gibt an, bei welcher Chassis- und Steckplatznummer der Fehler aufgetreten ist.
Adresse + 3	1. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des ersten Moduls, mit dem Daten ausgetauscht werden.
Adresse + 4	Daten vom 1. Modul lesen	Hier werden die aus dem ersten Modul gelesenen Daten abgelegt.
Adresse + 5	Daten für erstes Modul schreiben.	Dieser Datenwert wird zum ersten Modul geschrieben.
Adresse + 6	2. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des zweiten Moduls, mit dem Daten ausgetauscht werden.
Adresse + 7	Daten vom 2. Modul lesen	Hier werden die aus dem zweiten Modul gelesenen Daten abgelegt.
Adresse + 8	Daten für zweites Modul schreiben.	Dieser Datenwert wird zum zweiten Modul geschrieben.
Adresse + ((I-1) * 3) + 3	I. Chassis und Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des I-ten Moduls, mit dem Daten ausgetauscht werden.
Adresse + ((I-1) * 3) + 4	Daten vom I-ten Modul lesen.	Hier werden die aus dem I-ten Modul gelesenen Daten abgelegt.
Adresse + ((I-1) * 3) + 5	Daten für I-tes Modul schreiben.	Dieser Datenwert wird zum I-ten Modul geschrieben.
Adresse + ((N-1) * 3) + 3	Letztes Chassis und letzter Steckplatz	Chassis- und Steckplatznummer (in der Hexadezimalform RRSS, wobei RR die Chassisnummer und SS die Steckplatznummer darstellt) des letzten Moduls, mit dem Daten ausgetauscht werden.
Adresse + ((N-1) * 3) + 4	Daten vom letzten Modul lesen	Hier werden die aus dem letzten Modul gelesenen Daten abgelegt.
Adresse + ((N-1) * 3) + 5	Daten für letztes Modul schreiben.	Dieser Datenwert wird zum letzten Modul geschrieben.
Adresse + (N * 3) + 3	Ende der Listenanzeige.	Eine Null in diesem Wort zeigt das Ende der Modulliste an.

Fehlercodes

Wert	Beschreibung
1	Erfolg — Die Funktion wurde normal ausgeführt.
-1	Das Modul ist im angegebenen Steckplatz nicht vorhanden.
-2	Unpassendes Modul — Das Modul im angegebenen Steckplatz ist kein intelligentes Modul oder unterstützt nicht diese Funktionalität.
-3	Modul funktioniert nicht — Der Datenaustausch zwischen dem Modul im angegebenen Steckplatz und der CPU funktioniert nicht richtig.
-4	Paritätsfehler Lesedaten — Während einer Leseoperation von einem Erweiterungschassis oder dezentralen Chassis ist ein Paritätsfehler aufgetreten.
-5	Im Programmblock spezifizierte ungültige Funktion.

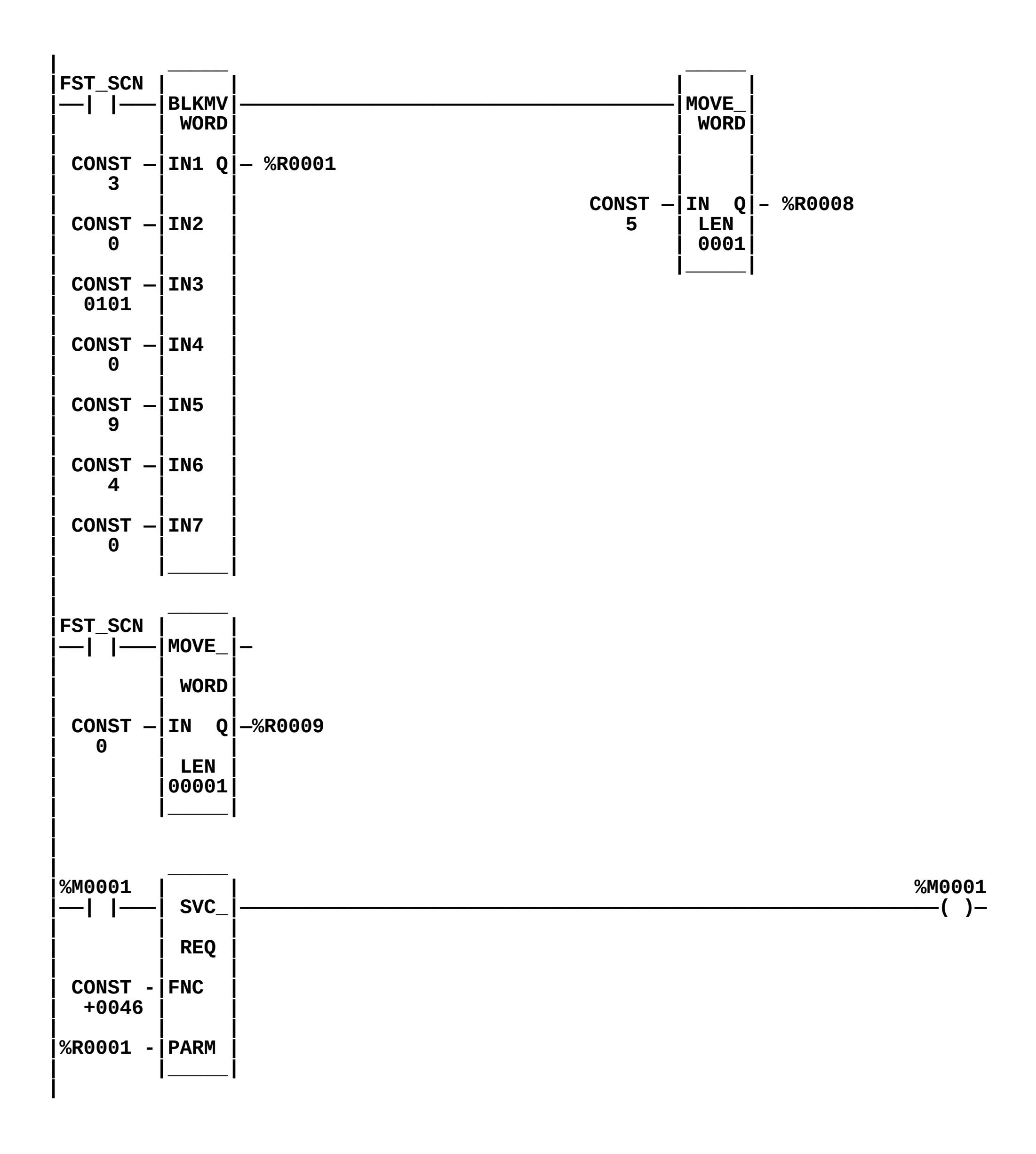
Beispiel 1

Das nachstehende Beispiel zeigt ein Lesen eines einzelnen Moduls in Chassis 2, Steckplatz 4. IN4 und IN5 müssen auf Null (0) gesetzt sein. IN6 und IN7 sind bei diesem Beispiel ohne Bedeutung. Wird die Funktion erfolgreich durchgeführt, stehen die Daten in %R0004.



Beispiel 2

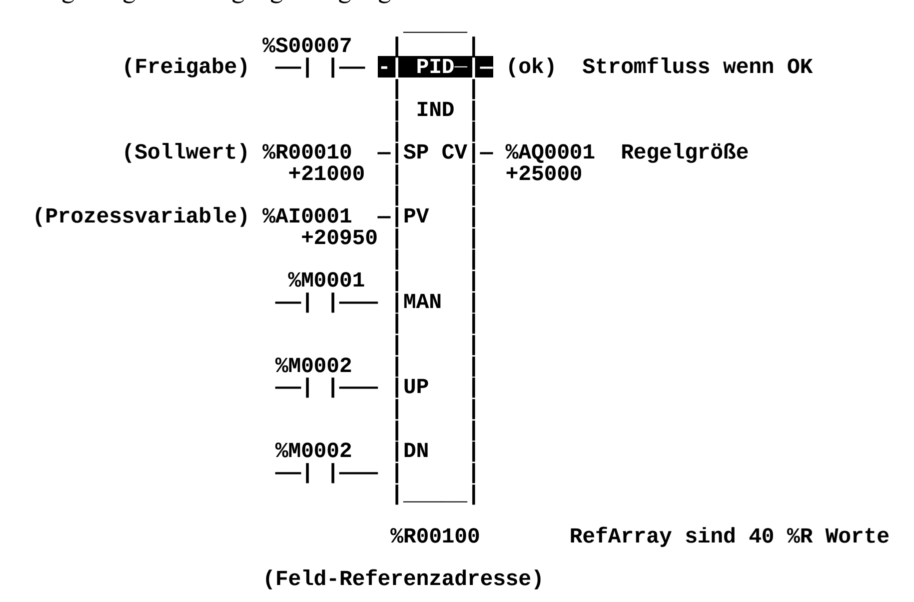
Bei diesem Beispiel werden die Statuszusatzdaten von dem Modul in Chassis 0, Steckplatz 4 und von dem Modul in Chassis 1, Steckplatz 1 gelesen und eine 5 in das erste Modul und eine 9 in das zweite Modul geschrieben. Beachten Sie, dass die Module nicht in der Reihenfolge der Steckplatznummern aufgeführt sein müssen. Die von dem Modul in Chassis 0, Steckplatz 4 gelesenen Daten werden in %R0007 eingetragen. Die von dem Modul in Chassis 1, Steckplatz 1 gelesenen Daten werden in %R0004 abgelegt.



PID

Die PID-Funktion (Proportional-Integral-Differential) ist der bekannteste allgemeine Algorithmus, der zur Prozessregelung benutzt wird. Der PID-Funktionsblock der Series 90 vergleicht den rückgekoppelten Wert einer Prozessvariablen mit einem Sollwert und aktualisiert den Ausgang der Stellgröße als Funktion des Fehlers.

Zur Lösung des PID-Algorithmus im gewünschten Zeitabschnitt verwendet der Block die PID-Regelverstärkung und weitere Parameter, die in einem Feld von 40 16-Bit-Worten (siehe Seite 12-73) gespeichert sind. Um kompatibel zu den 16-Bit-Analogprozessvariablen zu sein, sind alle Parameter ganzzahlige 16-Bit-Worte. Hierdurch kann der %AI-Speicher zur Eingabe von Prozessvariablen und der %AQ-Speicher zur Ausgabe von Regelgrößen verwendet werden. Das nachstehende Beispiel zeigt mögliche Eingangsbelegungen.



Als skalierte ganzzahlige 16-Bit-Zahlen müssen zahlreiche Parameter in PV-Zählwerten oder in CV-Zählwerten oder Einheiten definiert werden. Der SP-Eingang muss zum Beispiel über den gleichen Bereich wie PV skaliert werden, da der PID-Block die Abweichung durch eine Subtraktion dieser beiden Eingänge berechnet. In Übereinstimmung mit der Analogskalierung können die PV- und CV-Zählwerte –32000 oder 0 bis 32000 sein, oder, zur Anzeige von Variablen als 0,00% bis 100,00%, von 0 bis 10000. Die PV- und CV-Zählwerte brauchen nicht die gleiche Skalierung zu besitzen. In diesem Fall werden die Skalierungsfaktoren in den PID-Verstärkungen eingerechnet.

Hinweis

Die PID-Funktion kann höchstens einmal alle 10 Millisekunden ausgeführt werden. Dies kann zu einer Veränderung der gewünschten Ergebnisse führen, wenn Sie die Funktion in jedem Zyklus ausführen lassen und die Zyklusdauer kürzer als 10 Millisekunden ist. In einem solchen Fall wird die PID-Funktion erst ausgeführt, wenn so viele Zyklen durchlaufen wurden, dass eine Zeit von 10 Millisekunden aufgelaufen ist. Zum Beispiel: Bei einer Zykluszeit von 9 Millisekunden wird die PID-Funktion in jedem zweiten Zyklus mit einer aufgelaufenen Zeit von 18 Millisekunden pro Ausführung bearbeitet.

Parameter

Parameter	Beschreibung
Freigabe	Die PID-Funktion wird durchgeführt, wenn dieser Eingang über einen Kontakt aktiviert wird.
SP	SP ist der Regelsollwert. Wird er mit PV-Zahlen eingestellt, passt PID den Ausgangs-CV so an, dass PV zu SP passt (Nullfehler).
PV	Prozessvariableneingang vom geregelten Prozess (häufig ein %AI-Eingang).
MAN	Ist MAN (über einen Kontakt) stromführend, dann ist die PID-Funktion im **HANDBETRIEB**. Ist dieser Parameter nicht stromführend (0), ist der PID-Block in Automatikbetrieb.
UP	Der CV-Ausgang wird bei jeder Ausführung um 1 nach oben verändert, wenn dieser Eingang zusammen mit MAN aktiviert wird.*
DN	Der CV-Ausgang wird bei jeder Ausführung um 1 nach unten verändert, wenn dieser Eingang zusammen mit MAN aktiviert wird.*
RefArray Adresse	Die Adresse der PID-Steuerblockinformation (Anwenderparameter und interne Parameter). Benutzt 40 %R-Worte, die nicht gemeinsam genutzt werden können.
OK	Der OK-Ausgang wird durchgeschaltet, wenn die Funktion fehlerfrei ausgeführt wurde. Er ist abgeschaltet, wenn ein Fehler vorliegt.
CV	CV ist der Ausgang der Regelgröße zum Prozess (häufig ein analoger %AQ-Ausgang).

*Wird pro Zugriff auf die PID-Funktion um 1 erhöht (UP-Parameter) oder erniedrigt (DN-Parameter).

Zulässige Speichertypen

Parameter	Strom-fluss	%I	%Q	%M	%T	%S	%G	%R	%AI	%AQ	const	-
Freigabe	•											
SP		•	•	•	•		•	•	•	•	•	
PV		•	•	•	•		•	•	•	•		
MAN	•											
UP	•											
DN	•											
Adresse								•				
OK	•											•
CV		•	•	•	•		•	•	•	•		

• Zulässige Referenz oder Platz, an dem Strom durch die Funktion fließen kann.

PID-Parameterblock

Neben den beiden Eingangsworten und den 3 Handsteuerkontakten benutzt der PID-Block 13 der Parameter im Adressfeld RefArray. Diese Parameter müssen eingestellt werden, ehe der Block aufgerufen wird. Die restlichen Parameter werden von der SPS benutzt und sind nicht konfigurierbar. Die in nachstehender Tabelle aufgeführten %Ref ist die gleiche RefArray-Adresse wie die unten am PID-Block. Die Zahl nach dem Pluszeichen gibt den Offset im Feld an. Beginnt RefArray zum Beispiel bei %R100, enthält %R113 den Handbefehl, mit dem die Regelgröße und der Integrator im Handbetrieb eingestellt werden.

Tabelle 12-8. Übersicht PID-Parameter

Register	Parameter	Einheiten unteres Bit	Wertebereich
%Ref+0000	Reglernummer	Ganze Zahl	0 bis 255 (nur Anwenderanzeige)
%Ref+0001	Algorithmus	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0002	Ausführungsintervall	10 Millisekunden	0 (jeder Zyklus) bis 65535 (10,9 min). Für SPS 90-30 mind. auf 10 einstellen (s. Hinweis auf Seite 12-71).
%Ref+0003	Totzone +	PV-Zählwerte	0 bis 32000 (+ nie negativ)
%Ref+0004	Totzone —	PV-Zählwerte	–32000 bis 0 (nie positiv)
%Ref+0005	P-Verstärkung –Kp	0,01 CV%/PV%	0 bis 327,67 %/%
%Ref+0006	D-Verstärkung –Kd	0,01 Sekunden	0 bis 327,67 s
%Ref+0007	I-Zeit –Ki	Wiederholungen/1000 s	0 bis 32.767 Wiederholungen/s
%Ref+0008	CV-Anhebung/Ausgangsoffset	CV-Zählwerte	–32000 bis 32000 (zu Integratorausgang hinzuzählen)
%Ref+0009	Oberer Begrenzung	CV-Zählwerte	–32000 bis 32000(>%Ref+10) Ausgangsgrenze
%Ref+0010	Untere Begrenzung	CV-Zählwerte	–32000 bis 32000(<%Ref+09) Ausgangsgrenze
%Ref+0011	Nachführzeit	Sekunde/Vollausschlag	0 (keine) bis 32000 s bis zu 32000 CV
%Ref+0012	Konfigurationswort	Unterste 5 Bits benutzt	Bit 0 für 2 für Vorzeichenfehler, Ausgangspolarität, Diff.
%Ref+0013	Handbefehl	CV-Zählwerte	CV-Nachführung in Automatik- oder CV-Einstellung in Handbetrieb
%Ref+0014	Steuerwort	Von SPS verwaltet, wenn Bit 1 ***nicht*** gesetzt.	Sofern nicht anders eingestellt, von SPS unterhalten: Unteres Bit = 1 setzt Override (s. Beschreibung in Tabelle "Einzelheiten PID-Parameter" auf Seite 12-76)
%Ref+0015	Interner Sollwert	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0016	Interner CV	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0017	Interner PV	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar

Tabelle 12-8. Übersicht PID-Parameter - Fortsetzung

Register	Parameter	Einheiten unteres Bit	Wertebereich
%Ref+0018	Ausgang	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0019	Speicherung D-Größe	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0020 und %Ref+0021	Speicherung I-Größe	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0022	Speicherung Änderungsgröße	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0023	Uhr	--; wird von SPS gesetzt	
%Ref+0024		verwaltet	Nicht konfigurierbar
%Ref+0025	(letzte Ausführungszeit)		
%Ref+0026	Speicherung Y-Rest	--; wird von SPS gesetzt und verwaltet	Nicht konfigurierbar
%Ref+0027	Untergrenze für SP, PV	PV-Zählwerte	–32000 bis 32000 (>%Ref+28) für Anzeige
%Ref+0028	Obergrenze für SP, PV	PV-Zählwerte	–32000 bis 32000 (<%Ref+27) für Anzeige
%Ref+0029 +• %Ref+0034	Reserviert für interne Verwendung	--	Nicht konfigurierbar
%Ref+0035 • %Ref+0039	Reserviert für externe Verwendung	--	Nicht konfigurierbar

Für das Adressfeld RefArray müssen %R-Register in der SPS 90-30 verwendet werden. Beachten Sie, dass Sie für jeden Aufruf des PID-Blocks ein eigenes 40-Wort-Feld verwenden müssen, selbst wenn alle 13 Parameter jeweils gleich sind. Dies rührt daher, dass die anderen Worte in dem Feld für PID-interne Speichervorgänge verwendet werden. Stellen Sie sicher, dass das Feld nicht über das Speicherende hinausgeht.

Zur Konfiguration der Anwenderparameter wählen Sie die PID-Funktion und drücken dann **F10** , um ein Menü zu öffnen, in dem die Anwenderparameter angezeigt werden. Wählen Sie dann mit den Pfeiltasten die Felder aus und geben die entsprechenden Werte ein. Für die meisten Standardwerte können Sie 0 eingeben. Eine Ausnahme ist die obere CV-Begrenzung. Dieser Wert muss größer sein als die untere CV-Begrenzung, sonst kann der PID-Block nicht ablaufen. Beachten Sie, dass der PID-Block bei fehlerhaften Anwenderdaten **keinen** Stromfluss weitergibt. Verwenden Sie daher einen temporären Merker zur Überwachung, während Sie die Daten verändern.

Nachdem geeignete PID-Werte eingestellt wurden, sollten diese im BLKMOV als Konstanten definiert werden, so dass sie bei Bedarf zum erneuten Laden der PID-Standard-Anwenderparameter verwendet werden können.

Arbeitsweise der PID-Anweisung

Im normalen automatischen Betrieb wird der PID-Block in jedem Zyklus aufgerufen. Hierbei liegt am Freigabeeingang Stromfluss an und an den Handeingabekontakten liegt kein Stromfluss. Der Block vergleicht den aktuellen Wert der SPS-Betriebszeituhr mit der im internen RefArray gespeicherten Zeit, zu der der PID-Block zuletzt bearbeitet wurde. Ist diese Zeitdifferenz größer als das im dritten Wort (%Ref+2) definierte Ausführungsintervall, wird der PID-Algorithmus mit der Differenzzeit bearbeitet. Hierbei werden die letzte Ausführungszeit und der Regelgrößen-Ausgangswert aktualisiert. Im Automatikbetrieb wird der Regelgrößen-Ausgangswert im Handbefehlparameter %Ref+13 eingetragen.

Liegt an den Eingangskontakten für Freigabe und Handbetrieb Stromfluss an, wird der PID-Block in Handbetrieb versetzt und der Regelgrößen-Ausgangswert wird aus dem Handbefehl-Parameter %Ref+0013 eingestellt. Liegt an den Eingängen UP bzw. DN Stromfluss, wird bei jeder Bearbeitung des PID-Blocks das Handbetriebs-Befehlswort um einen CV-Zählwert erhöht oder erniedrigt. Um schnellere manuelle Änderungen der Regelgröße zu erzielen, kann auch jeder beliebige CV-Zählwert direkt zum Handbetriebs-Befehlswort addiert bzw. von ihm abgezogen werden.

Mit der oberen und unteren CV-Begrenzung grenzt der PID-Block den CV-Ausgangswert ein. Wird eine positive Nachführgrenze definiert, beschränkt dies die Geschwindigkeit, mit der sich der CV-Ausgangswert ändern kann. Werden entweder die CV-Amplitude oder die Änderungsgeschwindigkeit überschritten, wird der im Integrator gespeicherte Wert so eingestellt, dass CV an der Grenze liegt. Dieser Überschwingschutz (definiert auf Seite 12-78) bedeutet, dass sich der CV-Ausgangswert von der Begrenzung wegbewegt, sobald sich das Vorzeichen der Störgröße ändert, selbst wenn die Störgröße über einen längeren Zeitraum versucht hat, CV über (oder unter) die Begrenzung zu zwingen.

Durch diese Arbeitsweise, zusammen mit der Handbefehl-Nachführung von CV im Automatikbetrieb und der Einstellung von CV im Handbetrieb, wird ein stoßfreier Übergang zwischen Hand- und Automatikbetrieb gewährleistet. Die obere und untere CV-Begrenzung sowie die Nachführgrenze gelten für den CV auch im Handbetrieb, und der im Integrator gespeicherte interne Wert wird aktualisiert. Das heißt, dass bei einer Änderung des Handbefehls im Handbetrieb die CV-Ausgabe sich maximal mit der in der Nachführgrenze angegebenen Geschwindigkeit ändert und die obere bzw. untere Begrenzung nicht über- oder unterschreitet.

Hinweis

Eine bestimmte PID-Funktion darf nicht mehr als einmal pro Zyklus aufgerufen werden.

In der nachstehenden Tabelle finden Sie weitere Einzelheiten zu den in Tabelle 12-3 aufgeführten Parametern. Die bei den einzelnen Parametern in Klammern angegebenen Zahlen stellen den Offset im Datenfeld RefArray dar.

Tabelle 12-9.

Parameter	Beschreibung
Regelkreisnummer (00)	Die Verwendung der Reglernummer zur Identifizierung eines PID-Blocks erfolgt wahlweise. Die Nummer, eine ganze Zahl ohne Vorzeichen, die über eine Bedienschnittstelle definiert wird, bildet eine gemeinsame Identifizierung in der SPS. Die Reglernummer wird unterhalb der Blockadresse angezeigt, wenn das Programm von der Logicmaster 90-30/20/Micro Software aus überwacht wird.
Algorithmus (01)	Dieser Wert ist eine vorzeichenlose ganze Zahl, die von der SPS gesetzt wird und angibt, welcher Algorithmus von dem Funktionsblock verwendet wird. Der ISA-Algorithmus wird mit Algorithmus 1 bezeichnet, der unabhängige Algorithmus mit Algorithmus 2.
Ausführungsintervall (02)	Die Mindestzeit (in Schritten von 10 ms), die zwischen zwei Ausführungen des PID-Algorithmus liegen muss. Beispiel: Eingabe von 10 für ein Intervall von 100 ms. Wird 0 eingegeben, wird der Algorithmus bei jedem Aufruf des Blocks bearbeitet (siehe unten: Zeitlicher Ablauf der PID-Blöcke). Der PID-Algorithmus wird nur bearbeitet, wenn sich der Wert der Betriebszeituhr seit der letzten Bearbeitung mindestens um den hier angegebenen Zeitraum erhöht hat. Beachten Sie, dass die 90-30 keine Ausführungszeiten von weniger als 10 ms benutzt (siehe Hinweis auf Seite 12-71); bei kürzeren Zykluszeiten werden somit Zyklen übersprungen. Diese Funktion gleicht die seit der letzten Ausführung verstrichene Zeit mit einer Genauigkeit von 100 Mikrosekunden aus. Wird dieser Wert auf 0 gesetzt, wird die Funktion jedesmal ausgeführt, wenn sie freigegeben wird. Die weiter oben erwähnte 10-Millisekunden-Grenze gilt aber auch hier.
Totzone (+/—) (03/04)	INT-Werte, die die obere (+) und untere (–) Grenze der Totzone in PV-Zählwerten festlegen. Wird keine Totzone benötigt, dann sollte dieser Wert auf 0 gesetzt werden. Liegt die PID-Regelabweichung (SP – PV) bzw.(PV – SP) zwischen den Werten (+) und (–) der Totzone, wird die Funktion mit dem Fehlerwert 0 gelöst. Ist der Wert von Null verschieden, muss der Wert (+) größer als 0 und der Wert (–) kleiner als 0 sein, damit der PID-Block funktioniert. *Lassen Sie diese Werte auf 0, bis die PID-Regelverstärkung eingestellt oder verändert wird.* . Danach kann eine Totzone erforderlich werden zur Vermeidung kleiner Änderungen der CV-Ausgabe, die durch kleine Schwankungen der Regelabweichung (z.B. durch mechanischen Verschleiß) verursacht werden.
P-Verstärkung –Kp (05)	In der ISA-Version wird mit dieser INT-Zahl, dem Verstärkungsfaktor Kc, die Änderung von CV in CV-Zählwerten bei einer Änderung von 100 PV-Zählwerten in der Regelabweichung festgelegt. Der Wert wird angezeigt als 0,00 %/%, mit implizit zwei Dezimalstellen. Beispiel: Wird für Kp ein Wert 450 eingegeben, wird dieser Wert als 4,50 angezeigt und ergibt einen Beitrag zur PID-Ausgabe von Kp*Fehler/100 oder 450*Fehler/100. Bei der Einstellung eines PID-Regelkreises wird immer zuerst Kp eingestellt.
D-Verstärkung –Kd (06)	Mit dieser INT-Zahl wird festgelegt, wie sich CV in CV-Zählwerten ändert, wenn der Fehler oder PV sich um 1 PV-Zählwert pro 10 ms ändert. Als Zeitwert eingegeben, bei dem das untere Bit 10 ms angibt, wird der Wert angezeigt als 0,00 s, mit implizit zwei Dezimalstellen. Beispiel: Wird für Kp ein Wert 120 eingegeben, wird dieser Wert als 1,20 s angezeigt und ergibt einen Beitrag zur PID-Ausgabe von Kd * Fehlerdelta/Zeitdelta bzw., wenn sich der Fehler alle 30 ms um 4 PV-Zählwerte änderte, von 120*4/3. Mit Kd kann eine langsame Reaktion des Regelkreises beschleunigt werden; es ist allerdings sehr anfällig gegenüber Störungen des PV-Eingangssignals.
I-Zeit –Ki (07)	Mit dieser INT-Zahl wird festgelegt, wie sich CV in CV-Zählwerten ändert, wenn der Fehler konstant 1 PV-Zählwert wäre. Er wird angezeigt als 0,000 Wiederholungen/s mit implizit 3 Dezimalstellen. Beispiel: Wird für Ki ein Wert 1400 eingegeben, wird dieser Wert als 1,400 Wiederholungen/s angezeigt und ergibt einen Beitrag zur PID-Ausgabe von Ki*Fehler*dt oder, bei einem Fehler von 20 Zählwerten und einer SPS-Zykluszeit von 50 ms (Ausführungsintervall 0), 1400*20*50/1000 . Ki ist normalerweise nach Kp der zweite Verstärkungsparameter, der eingestellt wird.
CV-Anhebung/Ausgangs-Offset (08)	Dieser INT-Wert in CV-Zählwerten wird vor den Geschwindigkeits- und Amplitudenbegrenzungen zum PID-Ausgang addiert. Mit ihm können von Null verschiedene CV-Werte eingestellt werden, wenn nur Kp-Proportionalverstärkungen benutzt werden, bzw. zur Vorwärtsregelung dieses PID-Regelausgangs von einem anderen Regelkreis aus.

Tabelle 12-9. Einzelheiten zu PID-Parametern - Fortsetzung

Parameter	Beschreibung
Obere und untere CV-Begrenzung (09/10)	INT-Werte in CV-Zählwerten, die die höchsten und niedrigsten Werte von CV festlegen. Diese Werte müssen eingegeben werden; die obere Begrenzung muss positiver als die untere Begrenzung sein, da sonst der PID-Block nicht funktioniert. Mit diesen Werten werden normalerweise die auf physikalischen Grenzen beruhenden Grenzwerte des CV-Ausgangs festgelegt. Mit ihnen wird auch die Balkenanzeige von CV für LM90 oder ADS PID-Anzeige skaliert. Der Block besitzt einen Überschwingschutz, mit dem der Integratorwert verändert werden kann, wenn eine CV-Begrenzung erreicht ist.
Nachführgrenze (11)	Ein positiver UINT-Wert, mit dem angegeben wird, wie lange der CV-Ausgang mindestens brauchen muss, um von 0 auf 100% Vollausschlag oder 32000 CV-Zählwerte zu gelangen. Dies ist die umgekehrte Grenzgeschwindigkeit, mit der sich der CV-Ausgangswert ändern kann. Ist der Wert positiv, kann CV nicht schneller als mit 32000 CV-Zählwerten * Zeitdelta (s) / Nachführgrenze verändert werden. Beispiel: Bei einem Ausführungsintervall von 2,5 s und einer Nachführgrenze von 500 s, kann CV sich pro PID-Bearbeitung um maximal 32000*2,5/500 = 160 CV-Zählwerte ändern. Wie bei den CV-Begrenzungen gibt es einen Überschwingschutz, der den Integratorwert einstellt, wenn die CV-Grenzgeschwindigkeit überschritten wird. Bei einer Nachführgrenze von 0 gibt es keine CV-Grenzgeschwindigkeit. Beim Einstellen der PID-Schleifenverstärkung müssen Sie die Nachführgrenze auf 0 einstellen.
Konfigurationswort	Mit den untersten 5 Bits dieses Wortes werden drei PID-Standardeinstellungen verändert. Die übrigen Bits müssen auf 0 gesetzt werden. Setzen Sie das unterste Bit auf 1, um die Standard-PID-Regelabweichung von normalerweise (SP – PV) auf (PV – SP) abzuändern und dadurch das Vorzeichen der Rückführungsgröße zu ändern. Diese Funktion benötigen Sie für Abwärtsregelungen, bei denen CV mit ansteigendem PV reduziert werden muss. Setzen Sie das zweite Bit auf 1, um die Ausgangspolarität zu verändern, so dass CV am PID-Ausgang negativ und nicht wie normal positiv ist. Setzen Sie das vierte Bit auf 1, um den Differenzanteil so zu verändern, dass er die Änderung in der PV-Rückführungsgröße und nicht die normale Änderung in der Regelabweichung benutzt. Die unteren 5 Bits im Konfigurationswort werden nachstehend detailliert beschrieben: **Bit 0** = Fehlergröße. Ist dieses Bit auf 0 gesetzt, dann ist die Fehlergröße SP — PV. Ist das Bit 1, dann ist die Fehlergröße PV — SP. **Bit 1** = Ausgangspolarität. Ist dieses Bit 0, dann wird der CV-Ausgang auf den Ausgangswert der PID-Berechnung eingestellt. Ist dieses Bit 1, dann wird der CV-Ausgang auf den negativen Ausgangswert der PID-Berechnung eingestellt. **Bit 2** = Differentialwirkung auf PV. Ist dieses Bit 0, dann wirkt das Differentialverhalten auf die Fehlergröße. Ist dieses Bit 1, dann wirkt das Differentialverhalten auf PV. Alle anderen Bits müssen 0 sein. **Bit 3** = Wirkung der Totzone. Wird das Totzonenbit auf 0 gesetzt, ist die Totzone ohne Wir kung. Liegt die Regelabweichung innerhalb der Grenzen der Totzone, wird sie zwangsweise auf Null gesetzt. Im anderen Fall wird die Regelabweichung von den Grenzen der Totzone nicht beeinflusst. Wird das Totzonenbit auf 1 gesetzt, zeigt die Totzone Wirkung. Liegt die Regelabweichung innerhalb der Grenzen der Totzone, wird sie zwangsweise auf Null gesetzt. Liegt die Regelabweichung außerhalb dieser Grenzen, wird sie um die Totzonen-Grenzen reduziert (Regelabweichung = Regelabweichung – Totzonengrenze). **Bit 4** = Überschwingschutz. Ist dieses Bit 0, verwendet der Überschwingschutz eine Rückwärtsberechnung. Ist der Ausgang begrenzt, ersetzt dieses den akkumulativen Restwert von Y (definiert auf Seite 12-78) mit dem zur exakten Erzeugung des begrenzten Ausgangs erforderlichen Wert. Ist das Bit auf 1 gesetzt, wird der akkumulative Y-Wert mit dem Wert ersetzt, den Y zu Beginn der Berechnung hatte. In diesem Fall wird der Y-Wert vor der Begrenzung solange gehalten, wie der Ausgang begrenzt ist. **HINWEIS:** Der Überschwingschutz steht nur zur Verfügung bei 90-30 CPUs ab Aus gabestand 6.50. Denken Sie daran, dass die Bits in Potenzen von 2 gesetzt werden. Beispiel: Um das Konfigurationswort für die Standard-PID-Konfiguration auf 0 einzustellen, addieren Sie 1, um die Regelabweichung von SP–PV auf PV–SP zu verändern, oder Sie addieren 2, um die Ausgangspolarität von CV = PID auf CV = – PID abzuändern, oder Sie addieren 4, um die Differentialwirkung von Änderung Regelabweichung auf PV-Änderung umzuschalten, usw.

Tabelle 12-9. Einzelheiten zu PID-Parametern - Fortsetzung

Parameter	Beschreibung
Handbefehl (13)	Dieser INT-Wert wird im Automatikbetrieb auf den aktuellen CV-Ausgang gesetzt. Wird der Block auf Handbetrieb umgeschaltet, werden mit diesem Wert der CV-Wert und der interne Wert des Integrators innerhalb der oberen und unteren Begrenzung und der Nachführgrenzen eingestellt.
Steuerwort (14)	Dieser interne Parameter wird normalerweise auf 0 belassen. Wird das Overridebit auf 1 gesetzt, müssen dieses Wort und die anderen internen SP-, PV- und CV-Parameter für dezentralen Betrieb dieses PID-Blocks verwendet werden (siehe unten). Hierdurch können dezentrale Bedienerschnittstellen, wie zum Beispiel Computer, die Kontrolle vom SPS-Programm wegnehmen. Achtung: Wollen Sie dies verhindern, müssen Sie das Steuerwort auf 0 setzen. Ist das untere Bit 0, können die nächsten 4 Bits gelesen werden, um die Zustände der PID-Eingangskontakte solange zu verfolgen, wie am PID-Freigabeeingang Stromfluss anliegt. Diskrete Datenstruktur mit den ersten fünf Bitstellen im folgenden Format: **Bit: Wortwert: Funktion: Status oder ext. Aktion wenn Overridebit = 1:** 0 1 Override 0: Blockkontakte überwachen. 1: Blockkontakte extern einstellen 1 2 Hand/Auto 1: Block ist in **Handbetrieb**; andere Werte: Block ist in **Automatikbetrieb**. 2 4 Freigabe Muss normalerweise 1 sein, sonst wird der Block nie aufgerufen. 3 8 UP/höher 1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erhöht. 4 16 DN/niedr. 1 und Handbetrieb (Bit 1=1): CV wird bei jedem Durchlauf erniedrigt.
SP (15)	(Nicht konfigurierbar – von SPS eingestellt und verwaltet) Verfolgt SP-Eingang. Muss extern gesetzt werden, wenn Override = 1.
CV (16)	(Nicht konfigurierbar – von SPS eingestellt und verwaltet) Verfolgt CV-Ausgang.
PV (17)	(Nicht konfigurierbar – von SPS eingestellt und verwaltet) Verfolgt PV-Eingang. Muss extern gesetzt werden, wenn Overridebit = 1.
Ausgangswert (18)	(Nicht konfigurierbar – von SPS eingestellt und verwaltet) Dieses vorzeichenbehaftete Wort stellt den Ausgang des Funktionsblocks vor einer wahlweisen Invertierung dar. Wurde keine Ausgangskonvertierung konfiguriert und wurde das Ausgangspolaritätsbit im Steuerwort auf 0 gesetzt, ist dieser Wert gleich dem CV-Ausgang. Wurde Invertierung konfiguriert und wurde das Ausgangspolaritätsbit im Steuerwort auf 1 gesetzt, ist dieser Wert gleich dem negierten CV-Ausgang.
Speicherung D-Größe (19)	Intern zur Speicherung von Zwischenwerten verwendet. *Schreiben Sie nicht in diese Adresse.*
Speicherung I-Größe (20/21)	Intern zur Speicherung von Zwischenwerten verwendet. *Schreiben Sie nicht in diese Adresse.*
Speicherung Änderungsgröße (22)	Intern zur Speicherung von Zwischenwerten verwendet. *Schreiben Sie nicht in diese Adresse.*
Takt (23–25)	Interne Speicherung der Betriebszeit (letzte Ausführung des PID-Blocks). *Schreiben Sie nicht in diese Adressen.*
Y Rest (26)	Enthält den Rest für die Integrator-Teilungsskalierung bei stationärer Regelabweichung = 0.
Unterer und oberer Bereich (27/28)	Wahlweise INT-Werte in PV-Zählwerten, die den kleinsten und größten Anzeigewert bei horizontaler SP- und PV-Balkenanzeige über Logicmaster Zoom **Zoom** und ADS PID Frontplattenanzeige angeben.
Reserviert (29–34 und 35–39)	29–34 sind für interne Benutzung reserviert. 35–39 sind für externe Benutzung reserviert. Reserviert für Verwendung durch GE Fanuc. Sie dürfen nicht für andere Zwecke verwendet werden.

Interne Parameter im Adressfeld (RefArray)

Wie in Tabelle 12-3 auf den vorherigen Seiten beschrieben wird, liest der PID-Block 13 Anwenderparameter und verwendet den Rest des 40-Worte-Adressfeldes zur internen PID-Speicherung. Normalerweise brauchen Sie keinen dieser Werte zu verändern. Wenn Sie den PID-Block nach einer langen Verzögerung in Automatikbetrieb aufrufen, können Sie mit SVC_REQ #16 die aktuelle SPS-Betriebszeit in %Ref+23 laden, um die letzte PID-Bearbeitungszeit zu aktualisieren und damit einen Schrittwechsel am Integrator zu vermeiden. Haben Sie das Overridebit im Steuerwort (%Ref+14) auf 1 gesetzt, müssen die nächsten vier Bits des Steuerwortes zur Steuerung der Eingangskontakte des PID-Blocks gesetzt werden (siehe Tabelle 12-3 auf den vorherigen Seite). Die internen SP und PV müssen gesetzt werden, da Sie die Kontrolle über den PID-Block vom Kontaktplanprogramm weggenommen haben.

Einstellung von PID-Algorithmen (PIDISA oder PIDIND) und Verstärkungen

Bei der Programmierung des PID-Blocks kann entweder die unabhängige (PID_IND) oder die Standardversion des PID-Algorithmus gewählt werden. Der einzige Unterschied zwischen diesen Algorithmen besteht in der Art, in der die I- und D-Verstärkungen definiert sind. Um diesen Unterschied zu verstehen, müssen Sie folgende Punkte kennen:

Beide PID-Typen berechnen die Fehlergröße als SP – PV bzw. bei Rückwärtsregelung als PV – SP (wenn die Fehlergröße, d.h. das untere Bit 0 im Konfigurationswort %Ref+12, auf 1 gesetzt ist). Rückwärtsregelung kann eingesetzt werden, wenn sich die Änderungen des CV-Ausgangssignals und die des PV-Eingangssignals im Gegensatz zum Normalbetrieb (CV steigt wenn PV steigt) in unterschiedliche Richtungen bewegen sollen (CV fällt bei steigendem PV).

Regelabweichung = (SP – PV)　　　oder (PV – SP) wenn das untere Bit des Konfigurationswortes 1 ist.

Die Differentiation basiert normalerweise auf der Änderung der Regelabweichung seit der letzten PID-Bearbeitung. Hierdurch kann sich bei einer Änderung des SP-Wertes eine große Änderung der Ausgangsgröße ergeben. Ist dies nicht erwünscht, können Sie das dritte Bit des Konfigurationswortes auf 1 setzen, um den auf PV basierenden Differentialanteil zu berechnen. Das dt (Zeitdifferenz) wird bestimmt, indem der Zeitwert der letzten PID-Bearbeitung dieses Blocks von der aktuellen SPS-Betriebszeit abgezogen wird.

dt = aktuelle SPS-Betriebszeit – SPS-Betriebszeit bei der letzten Bearbeitung des PID-Blocks

Differentialanteil = (Regelabweichung – vorherige Regelabweichung)/dt oder (PV – vorheriger PV)/dt

wenn 3. Bit des Konfigurationswortes = 1

Der unabhängige PID-Algorithmus (PID_IND) berechnet die Ausgangsgröße als:

PID-Ausgang = Kp * Regelabweichung + Ki * Regelabweichung * dt + Kd * Differentialanteil + CV-Anhebung

Der Standard-ISA-Algorithmus (PID_ISA) sieht anders aus:

PID-Ausgang = Kc * (Regelabweichung + Regelabweichung * dt/Ti + Td * Differentialanteil) + CV-Anhebung

mit: Kc = Verstärkungsfaktor der Steuerung, Ti = Nachstellzeit, Td = Vorhaltezeit. Der Vorteil von ISA liegt darin, dass eine Veränderung von Kc den Anteil der I-, D- und P-Größen verändert, wo-

durch der Regelkreisabgleich einfacher wird. Haben Sie PID-Verstärkungen als Ausdrücke von Ti und Td, verwenden Sie

Kp = Kc   Ki = Kc/Ti   und   Kd = Kc/Td

um sie für die Verwendung als PID-Anwenderparametereingabe umzuwandeln.

Der Ausdruck CV-Anhebung ist eine additive Größe, die von den PID-Komponenten getrennt ist. Diese Größe kann benötigt werden, wenn Sie nur die Proportionalverstärkung Kp verwenden und wünschen, dass CV bei PV = SP und Regelabweichung = 0 von Null verschieden ist. In diesem Fall setzen Sie die CV-Anhebung auf den gewünschten CV-Wert, wenn PV am SP ist. CV-Anhebung können Sie auch zur Vorwärtsregelung einsetzen, wenn der CV-Ausgang dieses PID-Kreises über einen anderen PID-Kreis oder Steueralgorithmus eingestellt wird.

Wird eine Integrationsverstärkung Ki verwendet, wäre CV-Anhebung normalerweise 0, da der Integrator als automatische Anhebung arbeitet. Starten Sie im Handbetrieb, stellen den Integrator über das Hand-Befehlswort (%Ref+13) auf den gewünschten CV ein, und schalten dann um auf Automatikbetrieb. Dies funktioniert auch, wenn Ki = 0, nur wird hierbei nach dem Umschalten in Automatikbetrieb der Integrator nicht auf die Basis der Regelabweichung eingestellt.

Die nachstehende Abbildung zeigt die Funktionsweise des PID-Algorithmus:

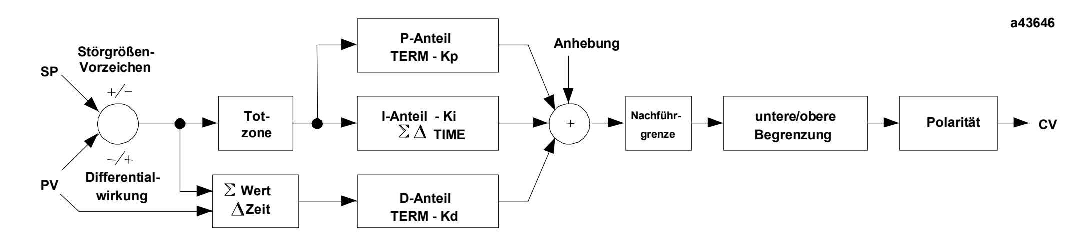


Abbildung 12-4. Unabhängiger PID-Algorithmus (PIDIND)

Der ISA-Algorithmus (PIDISA) ist ähnlich. Lediglich der Verstärkungsfaktor Kp hängt mit Ki und Kd zusammen, so dass sich die Integralverstärkung zu Kp * Ki und die Differentialverstärkung zu Kp * Kd ergibt. Regelabweichungs-Vorzeichen, Differentialwirkung und Polarität werde über Bits im Anwenderparameter "Konfigurationswort" eingestellt.

CV-Amplitude und Grenzgeschwindigkeiten

Der Block sendet den berechneten PID-Ausgangswert nicht direkt an CV. Beide PID-Algorithmen können der Ausgangs-Regelgröße Grenzen bezüglich Amplitude und Änderungsgeschwindigkeit auferlegen. Die maximale Änderungsgeschwindigkeit wird bestimmt, indem der maximale 100% CV-Wert (32000) durch die minimale Nachführzeit (falls >0 angegeben wurde) dividiert wird. Beispiel: Die Nachführzeit beträgt 100 Sekunden, die Grenzgeschwindigkeit ist dann 320 CV-Zählwerte/Sekunde. Mit einer dt-Bearbeitungszeit von 50 ms darf sich der neue CV-Ausgangswert um nicht mehr als 320*50/1000 oder 16 CV-Zählwerte vom vorherigen CV-Ausgangswert unterscheiden.

Der CV-Ausgangswert wird dann mit den oberen und unteren CV-Begrenzungen verglichen. Wird eine dieser Grenzen über- oder unterschritten, wird der CV-Ausgang auf den begrenzten Wert gesetzt. Werden bei der Veränderung von CV entweder Änderungs- oder Amplitudengrenzwert überschritten, wird der interne Integratorwert so angepasst, dass er dem Grenzwert entspricht, wodurch Überschwingen vermieden wird.

Zuletzt überprüft der Block die Ausgangspolarität (2. Bit des Konfigurationswortes %Ref+12) und ändert das Vorzeichen des Ausgangs, wenn das Bit 1 ist.

CV = begrenzter PID-Ausgang oder – begrenzter PID-Ausgang, wenn das Ausgangs-Polaritätsbit gesetzt ist

Ist der Block im Automatikmodus, wird der letzte CV im Handbefehl %Ref+13 eingetragen. Ist der Block im Handmodus, wird die PID-Gleichung übersprungen, da CV durch den Handbefehl eingestellt wird. Es werden jedoch alle Grenzwerte für Änderungsgeschwindigkeit und Amplitude überprüft. Das bedeutet, dass mit dem Handbefehl der Ausgangswert nicht über den oberen Grenzwert bzw. unter den unteren Grenzwert hinaus verändert werden kann. Außerdem kann sich der Ausgangswert nur mit der maximalen Änderungsgeschwindigkeit ändern.

Ausführungsintervall und zeitlicher Ablauf der PID-Blöcke

Der PID-Block ist eine digitale Realisierung einer analogen Steuerungsfunktion. Aus diesem Grund ist die Abtastzeit dt in der PID-Ausgangsgleichung nicht die bei Analogsteuerungen verfügbare unendlich kleine Zeitspanne. Die meisten der geregelten Prozesse können als eine Verstärkung mit einer Verzögerung erster oder zweiter Ordnung angenähert werden, möglicherweise mit einer reinen Zeitverzögerung. Der PID-Block stellt einen CV-Ausgangswert zum Prozess ein und benutzt den Rückkopplungs-PV vom Prozess, um den Fehler zu bestimmen, mit dem die nächste CV-Ausgabe angepasst wird. Die Gesamtzeitkonstante ist ein Schlüssel-Prozessparameter, der festlegt, wie schnell der PV reagiert, wenn sich der CV ändert. Wie im Abschnitt "Schleifenverstärkungs-Einstellung" weiter unten besprochen wird, ist die Gesamtzeitkonstante Tp+Tc bei einem System erster Ordnung die von PV benötigte Zeit, um bei einer sprunghaften Änderung von CV 63% des Endwertes zu erreichen. Der PID-Block kann einen Prozess nur steuern, wenn das Ausführungsintervall deutlich unter der halben Gesamtzeitkonstanten liegt. Größere Ausführungsintervalle machen den Vorgang instabil.

Das Ausführungsintervall sollte nicht größer sein als die Gesamtzeitkonstante geteilt durch 10 (oder, im schlimmsten Fall, herunter bis 5). Beispiel: Erreicht PV 2/3 seines Endwertes in 2 Sekunden, sollte das Ausführungsintervall kleiner als 0,2 Sekunden (oder maximal 0,4 Sekunden) sein. Auf der anderen Seite sollte das Ausführungsintervall nicht zu klein sein (wie zum Beispiel kleiner als die Gesamtzeitkonstante dividiert durch 1000), da sonst der Ausdruck Ki * Regelabweichung * dt für den PID-Integrator auf Null gerundet wird. Beispiel: Bei einem sehr langsamen Prozess, der 10 Stunden bzw. 36000 Sekunden benötigt, um den 63%-Wert zu erreichen, sollte das Ausführungsintervall auf 40 Sekunden oder mehr eingestellt werden.

Sofern der Prozess nicht gerade sehr schnell abläuft, braucht man kein Ausführungsintervall von 0 einzustellen, bei dem der PID-Algorithmus in jedem PID-Zyklus gelöst wird. Werden viele PID-Kreise verwendet, bei denen das Ausführungsintervall größer als die Zykluszeit ist, können sich bei der SPS-Zykluszeit große Schwankungen ergeben, wenn viele Kreise den Algorithmus gleichzeitig lösen. Eine einfache Lösung besteht darin, ein oder mehrere auf 1 gesetzte Bits durch ein Bitfeld von Nullen zu schicken, so dass dadurch Stromfluss zu einzelnen PID-Blöcken durchgeschaltet wird.

Festlegen der Prozesseigenschaften

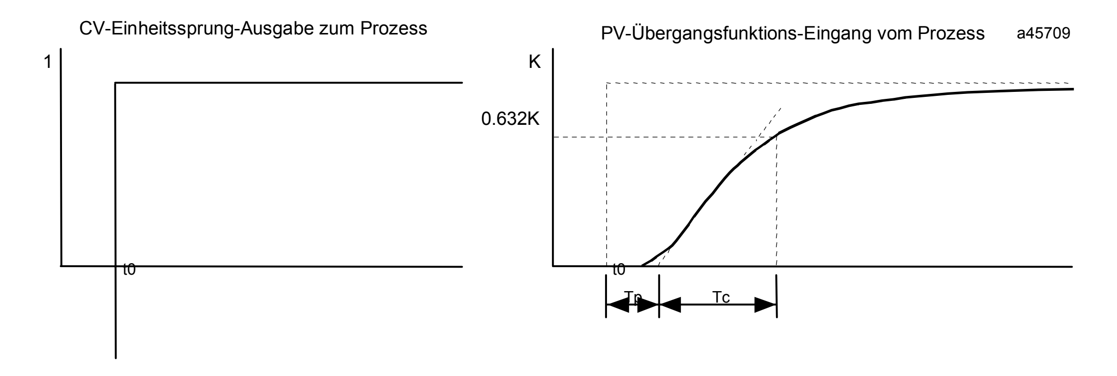
Die PID-Schleifenverstärkungen Kp, Ki und Kd werden von den Daten des gesteuerten Prozesses bestimmt. Beim Einstellen einer PID-Schleife stellen sich zwei Fragen:

1. Wie groß ist die Änderung von PV, wenn wir CV um einen festen Betrag ändern, bzw. wie groß ist die Leerlaufverstärkung?
2. Wie schnell reagiert das System, oder wie schnell ändert sich PV nach einer sprunghaften Änderung von CV?

Zahlreiche Prozesse können angenähert werden durch eine Prozessverstärkung, eine Verzögerung erster oder zweiter Ordnung, und eine reine Zeitverzögerung. Im Frequenzbereich ist die Übertragungsfunktion eines Verzögerungssystems zweiter Ordnung mit reiner Zeitverzögerung gegeben durch:

PV(s)/CV(s) = G(s) = K * e **(–Tp s)/(1 + Tc s)

Mit einer sprunghaften Änderung zum Zeitpunkt t0 im Zeitbereich ergibt sich folgende Änderung der Regelgröße:



Aus der PV-Übergangsfunktion können folgende Prozessmodellparameter bestimmt werden:

K	Leerlaufverstärkung = letzte Änderung von PV / Änderung von CV zum Zeitpunkt t0 (beachten Sie, dass K keinen Index hat).
Tp	Prozess- oder Pipeline-Zeitverzögerung oder Totzeit nach t0, bis sich der Prozessausgang PV anfängt zu bewegen.
Tc	Prozesszeitkonstante erster Ordnung; nach Tp von PV benötigte Zeit, um 63,2% des PV-Endwerts zu erreichen.

Normalerweise können Sie diese Parameter am schnellsten messen, indem Sie den PID-Block in Handmodus schalten und den CV-Ausgang um einen kleinen Schritt verändern. Hierzu verändern Sie den Handbefehl %Ref+13 und zeigen die PV-Antwort als Funktion der Zeit an. Bei langsamen Prozessen kann dies von Hand durchgeführt werden; bei schnelleren Prozessen sollten Sie einen Schreiber oder ein Computergraphik-Datenerfassungspaket verwenden. Die Sprunghöhe von CV sollte zwar groß genug sein, um eine merkliche Änderung von PV zu bewirken, aber nicht so groß, dass sie den gemessenen Prozess unterbrechen würde. Als Richtwert können Sie 2 bis 10% des Unterschieds zwischen oberer und unterer Begrenzung annehmen.

Einstellen der Anwenderparameter und Abgleich der Regelkreisverstärkungen

Da alle PID-Parameter vollständig vom geregelten Prozess abhängen, gibt es keine funktionierenden Standardwerte. Eine brauchbare Regelkreisverstärkung können Sie aber in folgenden Schritten iterativ ermitteln:

1. Setzen Sie zunächst alle Anwenderparameter auf 0. Setzen Sie dann die obere und untere CV-Begrenzung auf den höchsten bzw. niedrigsten CV-Wert, der zu erwarten ist. Stellen Sie das Ausführungsintervall auf die geschätzte Prozesszeitkonstante (oben) /10 bis 100.
2. Schalten Sie den Block auf Handbetrieb und stellen Sie Handbefehl (%Ref+13) auf unterschiedliche Werte ein. Prüfen Sie dabei, ob CV auf die obere oder untere Begrenzung gebracht werden kann. Schreiben Sie den PV an einem CV-Punkt auf und laden Sie diesen Wert in SP.
3. Stellen Sie Kp auf eine kleine Verstärkung (z.B. 100 * max. CV / max. PV) ein und schalten Sie den Handmodus ab. Verändern Sie SP sprunghaft auf 2 bis 10% des maximalen PV-Bereichs und beobachten Sie die Reaktion von PV. Erhöhen Sie Kp, wenn die PV-Sprungantwort zu langsam ausfällt bzw. verkleinern Sie Kp, wenn PV überschwingt und oszilliert, ohne einen stationären Wert anzunehmen.
4. Haben Sie Kp ermittelt, erhöhen Sie Ki so, dass ein Überschwingen auftritt, welches in 2 oder 3 Zyklen auf einen stationären Wert gedämpft wird. Es kann dabei erforderlich werden, Kp zu reduzieren. Probieren Sie es auch mit unterschiedlichen Sprunghöhen und CV-Arbeitspunkten.
5. Nachdem Sie passende Werte für Kp und Ki gefunden haben, addieren Sie versuchsweise Kd, um eine schnelle Reaktion auf Eingangsänderungen zu erhalten (vorausgesetzt, das System beginnt nicht zu schwingen). Kd wird häufig nicht benötigt und funktioniert nicht bei verrauschtem PV.
6. Überprüfen Sie die Verstärkungswerte über verschiedenen SP-Arbeitspunkten. Stellen Sie bei Bedarf zusätzlich Totzone und Nachführgrenze ein. Bei einigen Prozessen mit umgekehrter Wirkungsrichtung kann es erforderlich sein, die Vorzeichen- oder Polaritätsbits im Konfigurationswort zu setzen.

Regelkreisverstärkung einstellen—Abgleich nach Ziegler und Nichols

Nachdem Sie die drei Parameter K, Tp und Tc des Prozessmodells ermittelt haben, können Sie sie zur Abschätzung der PID-Anfangsregelkreisverstärkungen benutzen. Mit dem folgenden Verfahren, das Ziegler und Nichols in den 40ern entwickelt haben, kann eine gute Antwort auf Systemstörungen mit Verstärkungsfaktoren erzielt werden, die ein Amplitudenverhältnis von 1/4 erzeugen. Das Amplitudenverhältnis ist das Verhältnis des zweiten zum ersten Maximum im Regelkreis-Übertragungsverhalten.

1. Berechnen Sie die Reaktionsgeschwindigkeit:

   R = K/Tc

2. Berechnen Sie bei Proportionalregelung die Proportionalverstärkung Kp:

   Kp = 1/(R * Tp) = Tc/(K * Tp)

3. Berechnen Sie bei PI-Regelung:

   Kp = 0,9/(R * Tp) = 0,9 * Tc/(K * Tp)
   Ki = 0,3 * Kp/Tp

4. Berechnen Sie bei PID-Regelung:

   Kp = G/(R * Tp)   mit G von 1,2 bis 2,0
   Ki = 0,5 * Kp/Tp
   Kd = 0,5 * Kp * Tp

5. Überprüfen Sie, ob das Ausführungsintervall in diesem Bereich liegt: (Tp + Tc)/10 bis (Tp + Tc)/1000

Bei einem anderen Verfahren, der "idealen Abstimmung", erzielen Sie die besten Reaktionen auf Änderungen von SP, die nur von der Prozessverzögerung Tp oder der Totzeit verzögert werden.

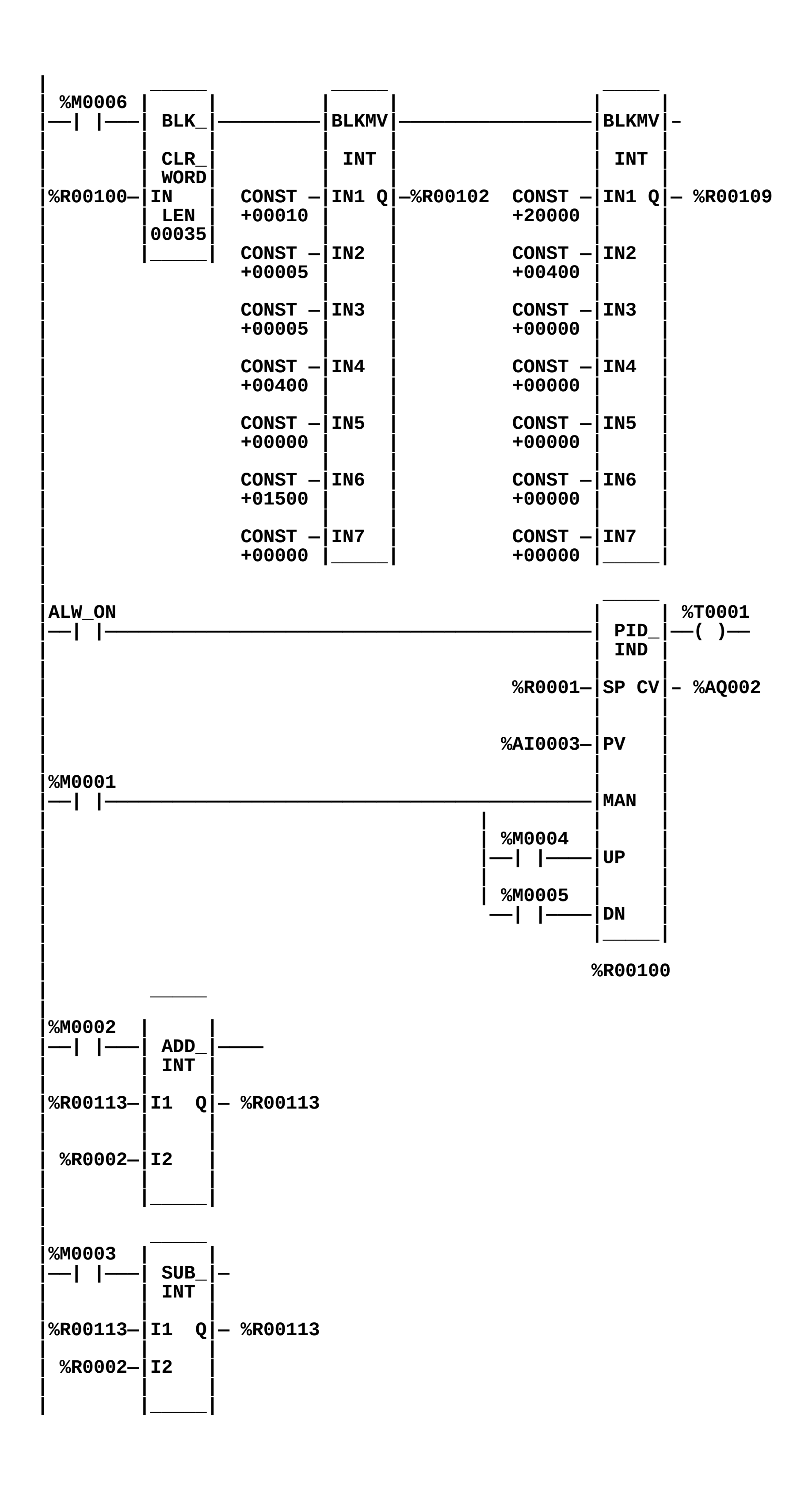
Kp = 2 * Tc/(3 * K * Tp)
Ki = Tc
Kd = Ki/4   wenn der Differentialanteil verwendet wird.

Nachdem die Anfangsverstärkungsfaktoren ermittelt wurden, müssen sie in ganzzahlige Anwenderparameter umgewandelt werden. Zur Vermeidung von Skalierungsproblemen sollte die Prozessverstärkung K als eine Änderung der Eingangs-PV-Zählwerte dividiert durch die sprungförmige Ausgangsänderung in CV-Zählwerten und nicht in technischen Prozess-PV- oder CV-Einheiten berechnet werden. Alle Werte sollten in Sekunden angegeben werden. Nachdem Kp, Ki und Kd bestimmt wurden, können Kp und Kd mit 100 multipliziert und als ganze Zahlen eingegeben werden. Ki kann mit 1000 multipliziert werden und in den Anwenderparameter %RefArray eingetragen werden.

Beispiel eines PID-Aufrufs

Das nachstehende Beispiel hat ein Ausführungsintervall von 100 ms, eine Kp-Verstärkung von 4,00 und eine Ki-Verstärkung von 1,500. Der Sollwert SP wird in %R1 gespeichert, während die Regelgröße CV in %AQ2 ausgegeben und die Prozessvariable in %AI3 zurückgeführt wird. Die oberen und unteren CV-Begrenzungen müssen in diesem Fall auf 20000 und 400 gesetzt werden; eine schmale Totzone von +5 und –5 wurde wahlweise eingesetzt. Das aus 40 Worten bestehende Adressfeld beginnt ab %R100. Normalerweise werden die Anwenderparameter im Adressfeld mit der PID-Zoomtaste **F10** eingestellt. Es kann jedoch auch %M6 gesetzt werden, um die 14 Worte ab %R102 (%Ref+2) aus den im Programm gespeicherten Konstanten neu zu initialisieren.

Der Block kann mit %M1 auf Handmodus umgeschaltet werden, so dass der Handbefehl %R113 eingestellt werden kann. Mit den Bits %M4 und %M5 können %R113, PID CV und Integrator in jeder 100-ms-Bearbeitung um 1 erhöht oder erniedrigt werden. Für schnelleren Handbetrieb kann mit den Bits %M2 und %M3 in jedem SPS-Zyklus der Wert aus %R2 zu %R113 addiert bzw. davon abgezogen werden. Der Ausgang %T1 ist durchgeschaltet, wenn die PID-Funktion in Ordnung ist.


%M0006
BLK_
CLR_
WORD
%R00100
IN
LEN
00035
BLKMV
INT
CONST
+00010
IN1 Q
%R00102
CONST
+00005
IN2
CONST
+00005
IN3
CONST
+00400
IN4
CONST
+00000
IN5
CONST
+01500
IN6
CONST
+00000
IN7
BLKMV
INT
CONST
+20000
IN1 Q
%R00109
CONST
+00400
IN2
CONST
+00000
IN3
CONST
+00000
IN4
CONST
+00000
IN5
CONST
+00000
IN6
CONST
+00000
IN7
ALW_ON
PID_
IND
%T0001
%R0001
SP CV
%AQ002
%AI0003
PV
%M0001
MAN
%M0004
UP
%M0005
DN
%R00100
%M0002
ADD_
INT
%R00113
I1 Q
%R00113
%R0002
I2
%M0003
SUB_
INT
%R00113
I1 Q
%R00113
%R0002
I2

Anhang A Befehlsausführungszeiten

Die speicherprogrammierbaren Steuerungen Series 90-30, 90-20 und Micro unterstützen zahlreiche unterschiedliche Funktionen und Funktionsblöcke. Die Tabellen in diesem Anhang zeigen für die einzelnen Funktionen den benötigten Speicherplatz sowie die Ausführungszeiten in Mikrosekunden. Die Speichergröße gibt an, wieviel Byte von der Funktion in einem Kontaktplan-Anwenderprogramm belegt werden.

Für jede Funktion werden zwei Ausführungszeiten angegeben:

Ausführungszeit	Beschreibung
Aktiv	Von einer Funktion bzw. einem Funktionsblock benötigte Zeit, wenn Stromfluss in die Funktion hinein und aus der Funktion heraus möglich ist. Die kürzesten Zeiten treten normalerweise auf, wenn der Block im Anwender-RAM steht (wortstrukturierter Speicher) und nicht im ISCP-Cache (bitorientierter Speicher).
Inaktiv	Für die Ausführung einer Funktion benötigte Zeit, wenn zwar Stromfluss in die Funktion oder den Funktionsblock möglich ist, diese aber in einem inaktiven Zustand sind (z.B. ein Zeitgeber, der im Rücksetzzustand gehalten wird).

Hinweis

Zeitgeber und Zähler werden immer aktualisiert, wenn sie in der Logik angetroffen werden; Zeitgeber um den seit dem letzten Zyklus aufgelaufenen Zeitwert und Zähler um einen Zählwert.

Hinweis

Bei den SPS CPUs 350, 351, 352 und 360 sind die Zeiten mit Ausnahme der MOVE-Funktion identisch. Diese Funktion ist bei der CPU 350 anders — siehe Hinweis unter der Tabelle auf Seite A-6.

Tabelle A-1. Befehlsausführungszeiten, Standardmodelle

Funktions-gruppe	Funktion	Freigegeben				Inaktiv				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Zeitgeber	Einschaltverzögerung	146	81	80	42	105	39	38	21	–	–	–	–	15
	Abschaltverzögerung	98	47	44	23	116	63	58	32	–	–	–	–	9
	Zeitgeber	122	76	75	40	103	54	53	30	–	–	–	–	15
Zähler	Aufwärtszähler	137	70	69	36	130	63	62	33	–	–	–	–	11
	Abwärtszähler	136	70	69	37	127	61	61	31	–	–	–	–	11
Arithmetik funktionen	Addition (INT)	76	47	46	24	41	0	1	0	–	–	–	–	13
	Addition (DINT)	90	60	60	34	41	1	0	0	–	–	–	–	13
	Subtraktion (INT)	75	46	45	25	41	0	1	0	–	–	–	–	13
	Subtraktion (DINT)	92	62	62	34	41	1	0	0	–	–	–	–	13
	Multiplikation (INT)	79	49	50	28	41	0	1	0	–	–	–	–	13
	Multiplikation (DINT)	108	80	101	43	41	1	0	0	–	–	–	–	13
	Division (INT)	79	51	50	27	41	0	1	0	–	–	–	–	13
	Division (DINT)	375	346	348	175	41	1	0	0	–	–	–	–	13
	Modulo Division (INT)	78	51	49	27	41	0	1	0	–	–	–	–	13
	Modulo Division (DINT)	134	103	107	54	41	1	0	0	–	–	–	–	13
	Quadratwurzel (INT)	153	124	123	65	42	0	1	0	–	–	–	–	9
	Quadratwurzel (DINT)	268	239	241	120	42	0	0	1	–	–	–	–	9
Relationale Funktionen	Gleich (INT)	66	35	36	19	41	1	1	0	–	–	–	–	9
	Gleich (DINT)	86	56	54	29	41	1	0	0	–	–	–	–	9
	Ungleich (INT)	67	39	35	22	41	1	1	0	–	–	–	–	9
	Ungleich (DINT)	81	51	51	28	41	1	0	0	–	–	–	–	9
	Größer als (INT)	64	33	35	20	41	1	1	0	–	–	–	–	9
	Größer als (DINT)	89	59	58	32	41	1	0	0	–	–	–	–	9
	Größer als/gleich (INT)	64	36	34	19	41	1	1	0	–	–	–	–	9
	Größer als/gleich (DINT)	87	58	57	30	41	1	0	0	–	–	–	–	9
	Kleiner als (INT)	66	35		19	41	1	1	0	–	–	–	–	9
	Kleiner als (DINT)	87	57		30	41	1	1	0	–	–	–	–	9
	Kleiner als/gleich (INT)	66	36	34	21	41	1	1	0	–	–	–	–	9
	Kleiner als/gleich (DINT)	86	57	56	31	41	1	1	0	–	–	–	–	9
	Bereich (INT)	92	58	54	29	46	1	0	1	–	–	–	–	15
	Bereich (DINT)	106	75	57	37	45	0	0	0	–	–	–	–	15
	Bereich (WORD)	93	60	54	29	0	0	0	0	–	–	–	–	15

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5.0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 340 und 341 (Ausgabestand 7 für 331).
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

Tabelle A-1. Befehlsausführungszeiten, Standardmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben				Inaktiv				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Bitoperationen	Logisch UND	67	37	37	22	42	0	0	1	–	–	–	–	13
	Logisch ODER	68	38	38	21	42	0	0	1	–	–	–	–	13
	Logisches Exklusiv-ODER	66	38	37	20	42	0	1	1	–	–	–	–	13
	Logisch invertiert, NOT	62	32	31	17	42	0	1	1	–	–	–	–	9
	Bit nach links verschieben	139	89	90	47	74	26	23	13	11,61	11,61	12,04	6,29	15
	Bit nach rechts verschieben	135	87	85	45	75	26	24	13	11,63	11,62	12,02	6,33	15
	Bit nach links rotieren	156	127	126	65	42	1	1	0	11,70	11,78	12,17	6,33	15
	Bit nach rechts rotieren	146	116	116	62	42	1	1	0	11,74	11,74	12,13	6,27	15
	Bitposition	102	72	49	38	42	1	0	0	–	–	–	–	13
	Bit auf 0 setzen	68	38	35	21	42	1	1	1	–	–	–	–	13
	Bit testen	79	49	51	28	41	0	0	1	–	–	–	–	13
	Bit auf 1 setzen	67	37	37	20	42	0	0	0	–	–	–	–	13
	Vergleich mit Maske (WORD)	217	154	141	74	107	44	39	21	–	–	–	–	25
	Vergleich mit Maske (DWORD)	232	169	156	83	108	44	39	22	–	–	–	–	25
Daten kopieren	Kopieren (INT)	68	37	39	20	43	0	0	0	1,62	1,62	5,25	1,31	13
	Kopieren (BIT)	94	62	64	35	42	0	0	0	12,61	12,64	12,59	6,33	13
	Kopieren (WORD)	67	37	40	20	41	0	0	0	1,62	1,63	5,25	1,31	13
	Block kopieren (INT)	76	48	50	28	59	30	30	16	–	–	–	–	27
	Block kopieren (WORD)	76	48	49	29	59	29	28	15	–	–	–	–	27
	Block löschen	56	28	27	14	43	0	0	0	1,35	1,29	1,40	0,78	9
	Schieberegister (BIT)	201	153	153	79	85	36	34	18	0,69	0,68	0,71	0,37	15
	Schieberegister (WORD)	103	53	52	29	73	25	23	12	1,62	1,62	2,03	1,31	15
	Bitfolgesteuerung	165	101	99	53	96	31	29	16	0,07	0,07	0,08	0,05	15
	COMM_REQ	1317	1272	1489	884	41	2	0	0	–	–	–	–	13
Tabelle	Feld kopieren													
	INT	230	201	177	104	72	41	40	20	1,29	1,15	10,56	2,06	21
	DINT	231	202	181	105	74	44	42	23	3,24	3,24	10,53	2,61	21
	BIT	290	261	229	135	74	43	42	23	–0,03	–0,03	-0,01	0,79	21
	BYTE	228	198	176	104	74	42	42	23	0,81	0,82	8,51	1,25	21
	WORD	230	201	177	104	72	41	40	20	1,29	1,15	10,56	2,06	21
	Suche gleiche													
	INT	197	158	123	82	78	39	37	20	1,93	1,97	2,55	1,55	19
	DINT	206	166	135	87	79	38	36	21	4,33	4,34	4,55	2,44	19
	BYTE	179	141	117	74	78	38	36	21	1,53	1,49	1,83	1,03	19
	WORD	197	158	123	82	78	39	37	20	1,93	1,97	2,55	1,55	19

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5.0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 340 und 341 (Ausgabestand 7 für 331).
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-1. Befehlsausführungszeiten, Standardmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben				Inaktiv				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
	Suche ungleiche													
	INT	198	159	124	83	79	39	36	21	1,93	1,93	2,48	1,52	19
	DINT	201	163	132	84	79	37	35	21	6,49	6,47	6,88	3,82	19
	BYTE	179	141	117	73	79	38	36	19	1,54	1,51	1,85	1.05	19
	WORD	198	159	124	83	79	39	36	21	1,93	1,93	2,48	1,52	19
	Suche größer als													
	INT	198	160	125	82	79	37	38	19	3,83	3,83	4,41	2,59	19
	DINT	206	167	135	88	78	38	36	20	8,61	8,61	9,03	4,88	19
	BYTE	181	143	118	73	79	37	36	19	3,44	3,44	3,75	2,03	19
	WORD	198	160	125	82	79	37	38	19	3,83	3,83	4,41	2,59	19
	Suche größer als/gleich													
	INT	197	160	124	83	77	38	36	20	3,86	3,83	4,45	2,52	19
	DINT	205	167	136	87	80	39	36	21	8,62	8,61	9,02	4,87	19
	BYTE	180	142	118	75	79	37	37	20	3,47	3,44	3,73	2,00	19
	WORD	197	160	124	83	77	38	36	20	3,86	3,83	4,45	2,52	19
	Suche kleiner als													
	INT	199	159	124	84	78	38	36	20	3,83	3,86	4,48	2,48	19
	DINT	206	168	135	87	79	38	38	19	8,62	8,60	-1,36	4,88	19
	BYTE	181	143	119	75	80	38	37	20	3,44	3,44	3,75	2,00	19
	WORD	199	159	124	84	78	38	36	20	3,83	3,86	4,45	2,48	19
	Suche kleiner als/gleich													
	INT	200	158	124	82	79	38	37	21	3,79	3,90	4,45	2,55	19
	DINT	207	167	137	88	78	39	37	19	8,60	8,61	9,01	4,86	19
	BYTE	180	143	119	74	78	40	37	19	3,46	3,44	3,73	2,02	19
	WORD	200	158	124	82	79	38	37	21	3,79	3,90	4,45	2,55	19
Umwandlung	Umwandlung in INT	74	46	39	25	42	1	1	1	–	–	–	–	9
	Umwandlung in BCD–4	77	50	34	25	42	1	1	1	–	–	–	–	9

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5.0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 340 und 341 (Ausgabestand 7 für 331).
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-1. Befehlsausführungszeiten, Standardmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben				Inaktiv				Inkrement				Größe
		311	313	331	340/41	311	313	331	340/41	311	313	331	340/41	
Bedienelement	Unterprogrammaufruf	155	93	192	85	41	0	0	0	–	–	–	–	7
	E/A-Aktualisierung	309	278	323	177	38	1	0	0	–	–	–	–	12
	PID – ISA-Algorithmus	1870	1827	1812	929	91	56	82	30	–	–	–	–	15
	PID – IND-Algorithmus	2047	2007	2002	1017	91	56	82	30	–	–	–	–	15
	Ende-Anweisung	–	–	–	–	–	–	–	–	–	–	–	–	–
	Bedienanforderung													
	#6	93	54	63	45	41	2	0	0	–	–	–	–	9
	#7 (lesen)	–	37	309	161	–	2	0	0	–	–	–	–	9
	#7 (setzen)	–	37	309	161	–	2	0	0	–	–	–	–	9
	#14	447	418	483	244	41	2	0	0	–	–	–	–	9
	#15	281	243	165	139	41	2	0	0	–	–	–	–	9
	#16	131	104	115	69	41	2	0	0	–	–	–	–	9
	#18	–	56	300	180	–	2	0	0	–	–	–	–	9
	#23	1689	1663	1591	939	43	1	0	0	–	–	–	–	9
	#26//30*	1268	1354	6680	3538	42	0	0	0	–	–	–	–	9
	#29	–	–	55	41	–	–	1	0	–	–	–	–	9
	Geschachtelte MCR/ENDMCR-Kombination	135	73	68	39	75	25	21	12	–	–	–	–	8

*Bedienanforderung #26/30 wurde mit schnellem Zähler und 16-Punkte-Ausgangsmodul in Chassis mit 5 Steckplätzen gemessen.

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 5.0 der Logicmaster 90-30/20 Software für CPU-Modelle 311, 313, 340 und 341 (Ausgabestand 7 für 331).
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-2. Befehlsausführungszeiten, Hochleistungsmodelle

Funktions-gruppe	Funktion	Freigegeben 350/351/36x	Inaktiv 350/351/36x	Inkrement 350/351/36x	Freigeg. 352	Inaktiv 352	Inkrement 352	Grö-ße
Zeitgeber	Einschaltverzögerung	4	6	–	4	5	–	15
	Zeitgeber	3	3	–	2	2	–	15
	Abschaltverzögerung	3	3	–	3	2	–	15
Zähler	Aufwärtszähler	1	3	–	2	2	–	13
	Abwärtszähler	3	3	–	1	2	–	13
Arithmetik-funktionen	Addition (INT)	2	0	–	1	0	–	13
	Addition (DINT)	2	0	–	2	0	–	19
	Addition (REAL)	52	0	–	33	0	–	17
	Subtraktion (INT)	2	0	–	1	0	–	13
	Subtraktion (DINT)	2	0	–	2	0	–	19
	Subtraktion (REAL)	53	0	–	34	0	–	17
	Multiplikation (INT)	21	0	–	21	0	–	13
	Multiplikation (DINT)	24	0	–	24	0	–	19
	Multiplikation (REAL)	68	1	–	38	1	–	17
	Division (INT)	22	0	–	22	0	–	13
	Division (DINT),	25	0	–	25	0	–	19
	Division (REAL)	82	2	–	36	2	–	17
	Modulo Division (INT)	21	0	–	21	0	–	13
	Modulo Division (DINT)	25	0	–	25	0	–	19
	Quadratwurzel (INT)	42	1	–	41	1	–	10
	Quadratwurzel (DINT)	70	0	–	70	0	–	13
	Quadratwurzel (REAL)	137	0	–	35	0	–	11
Trigonometrische Funktionen	SIN (REAL)	360	0	–	32	0	–	11
	COS (REAL)	319	0	–	29	0	–	11
	TAN (REAL)	510	1	–	32	1	–	11
	ASIN (REAL)	440	0	–	45	0	–	11
	ACOS (REAL)	683	0	–	63	0	–	11
	ATAN (REAL)	264	1	–	33	1	–	11
Logarithmische Funktionen	LOG (REAL)	469	0	–	32	0	–	11
	LN (REAL)	437	0	–	32	0	–	11
Exponentialwert	EXP	639	0	–	42	0	–	11
	EXPT	89	1	–	54	1	–	17
Bogenmaß-Umrechnung	Umwandeln RAD in DEG	65	1	–	32	1	–	11
	Umwandeln DEG in RAD	59	0		32	0		11

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.

Tabelle A-2. Befehlsausführungszeiten, Hochleistungsmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben 350/351/36x	Inaktiv 350/351/36x	Inkrement 350/351/36x	Freigeg. 352	Inaktiv 352	Inkrement 352	Größe
Relationale Funktionen	Gleich (INT)	1	0	–	1	0	–	10
	Gleich (DINT)	2	0	–	2	0	–	16
	Gleich (REAL)	57	0	–	28	0	–	14
	Ungleich (INT)	1	0	–	1	0	–	10
	Ungleich (DINT)	1	0	–	1	0	–	16
	Ungleich (REAL)	62	0	–	31	0	–	14
	Größer als (INT)	1	0	–	1	0	–	10
	Größer als (DINT)	1	0	–	1	0	–	16
	Größer als (REAL)	57	0	–	32	0	–	14
	Größer als/gleich (INT)	1	0	–	1	0	–	10
	Größer als/gleich (DINT)	1	0	–	1	0	–	10
	Größer als/gleich (REAL)	57	1	–	31	1	–	14
	Kleiner als (INT)	1	0	–	1	0	–	10
	Kleiner als (DINT)	1	0	–	1	0	–	16
	Kleiner als (REAL)	58	1	–	36	1	–	14
	Kleiner als/gleich (INT)	1	0	–	1	0	–	10
	Kleiner als/gleich (DINT)	3	0	–	3	0	–	16
	Kleiner als/gleich (REAL)	37	0	–	37	0	–	14
	Bereich (INT)	2	1	–	2	1	–	13
	Bereich (DINT)	2	1	–	2	1	–	22
	Bereich (WORD)	1	0	–	1	0	–	13
Bit-operation	Logisch UND	2	0	–	2	0	–	13
	Logisch ODER	2	0	–	2	0	–	13
	Logisches Exklusiv-ODER	1	0	–	1	0	–	13
	Logisch invertiert, NOT	1	0	–	1	0	–	10
	Bit nach links verschieben	31	1	1,37	31	1	1,37	16
	Bit nach rechts verschieben	28	0	3,03	28	0	3,03	16
	Bit nach links rotieren	25	0	3,12	25	0	3,12	16
	Bit nach rechts rotieren	25	0	4,14	25	0	4,14	16
	Bitposition	20	1	–	20	1	–	13
	Bit auf 0 setzen	20	0	–	20	0	–	13
	Bit testen	20	0	–	20	0	–	13
	Bit auf 1 setzen	19	1	–	19	1	–	13
	Vergleich mit Maske (WORD)	52	0	–	52	0	–	25
	Vergleich mit Maske (DWORD)	50	0	–	49	0	–	25

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 351 und 352.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-2. Befehlsausführungszeiten, Hochleistungsmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben 350/351/36X	Inaktiv 350/351/36X	Inkrement 350/351/36X	Freigeg. 352	Inaktiv 352	Inkrement 352	Größe
Daten kopieren	Kopieren (INT)	2	0	0,41	2	0	0,41	10
	Kopieren (BIT)	28	0	4,98	28	0	4,98	13
	Kopieren (WORD)	2	0	0,41	2	0	0,41	10
	Kopieren (REAL)	24	1	0,82	24	1	0,82	13
	Block kopieren (INT)	2	0	–	2	0	–	28
	Block kopieren (WORD)	4	4	–	3	0	–	28
	Block kopieren (REAL)	41	0	–	41	0	–	13
	Block löschen	1	0	0,24	1	0	0,24	11
	Schieberegister (BIT)	49	0	0,23	46	0	0,23	16
	Schieberegister (WORD)	27	0	0,41	27	0	0,41	16
	Bitfolgesteuerung	38	22	0,02	38	22	0,02	16
	COMM_REQ	765	0	–	765	0	–	13
Tabelle	Feld kopieren							
	INT	54	0	0,97	54	0	0,97	22
	DINT	54	0	0,81	54	0	0,81	22
	BIT	69	0	0,37	69	0	0,36	22
	BYTE	54	1	0,64	54	1	0,64	22
	WORD	54	0	0,97	54	0	0,97	22
	Suche gleiche							
	INT	37	0	0,62	37	0	0,62	19
	DINT	41	1	1,38	41	1	1,38	22
	BYTE	35	0	0,46	35	0	0,46	19
	WORD	37	0	0,62	37	0	0,62	19
	Suche ungleiche							
	INT	37	0	0,62	37	0	0,62	19
	DINT	38	0	2,14	38	0	2,14	22
	BYTE	37	0	0,47	37	0	0,47	19
	WORD	37	0	0,62	37	0	0,62	19
	Suche größer als							
	INT	37	0	1,52	37	0	1,52	19
	DINT	39	0	2,26	39	0	2,26	22
	BYTE	36	1	1,24	36	1	1,24	19
	WORD	37	0	1,52	37	0	1,52	19
	Suche größer als/gleich							
	INT	37	0	1,48	37	0	1,48	19
	DINT	39	0	2,33	39	0	2,33	22
	BYTE	37	1	1,34	37	1	1,34	19
	WORD	37	0	1,48	37	0	1,48	19

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 350 und 360.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-2. Befehlsausführungszeiten, Hochleistungsmodelle - Fortsetzung

Funktions-gruppe	Funktion	Freigegeben 350/351/36x	Inaktiv 350/351/36x	Inkrement 350/351/36x	Freigeg. 352	Inaktiv 352	Inkrement 352	Größe
	Suche kleiner als							
	INT	37	0	1,52	37	0	1,52	19
	DINT	41	1	2,27	41	1	2,27	22
	BYTE	37	0	1,41	37	0	1,41	19
	WORD	37	0	1,52	37	0	1,52	19
	Suche kleiner als/gleich							
	INT	38	0	1,48	38	0	1,48	19
	DINT	40	1	2,30	40	1	2,30	22
	BYTE	37	0	1,24	37	0	1,24	19
	WORD	38	0	1,48	38	0	1,48	19
Umwand-lung	Umwandlung in INT	19	1	–	19	1	–	10
	Umwandlung in BCD-4	21	1	–	21	1	–	10
	Umwandlung in REAL	27	0	–	21	0	–	8
	Umwandlung in WORD	28	1	–	30	1	–	11
	Abschneiden auf INT	32	0	–	32	0	–	11
	Abschneiden auf DINT	63	0	–	31	0	–	11
Steuerung	Unterprogrammaufruf	72	1	–	73	1	–	7
	E/A-Aktualisierung	114	1	–	115	1	–	13
	PID – ISA-Algorithmus*	162	34	–	162	34	–	16
	PID – IND-Algorithmus*	146	34	–	146	34	–	16
	Ende-Anweisung	–	–	–	–	–	–	–
	Bedienanforderung							
	#6	22	1	–	22	1	–	10
	#7 (lesen)	75	1	–	75	1	–	10
	#7 (setzen)	75	1	–	75	1	–	10
	#14	121	1	–	121	1	–	10
	#15	46	1	–	46	1	–	10
	#16	36	1	–	36	1	–	10
	#18	261	1	–	261	1	–	10
	#23	426	0	–	426	0	–	10
	#26//30**	2260	1	–	2260	1	–	10
	#29	20	0	–	20	0	–	10
	#43							
	Geschachtelte MCR/ENDMCR-Kombination	1	1	–	1	1	–	4
	Ablaufaufzeichnung (SER)	Siehe Tabelle A-3	26,50	Siehe Tabelle A-3				

*Die PID-Zeiten sind angegeben auf der Grundlage von Ausgabestand 6.5 der CPU 351.
*Bedienanforderung #26/30 wurde mit schnellem Zähler und 16-Punkte-Ausgangsmodul in Chassis mit 5 Steckplätzen gemessen.

Hinweise:

1. Die Zeit (in Mikrosekunden) ist angegeben auf der Grundlage von Ausgabestand 7 der Logicmaster 90-30/20 Software für CPU-Modelle 350 und 360.
2. Bei den Tabellenfunktionen ist das Inkrement in Einheiten der angegebenen Länge; bei Bitoperationen in Mikrosekunden/Bit; bei Datenverschiebefunktionen in Mikrosekunden/Anzahl Bits oder Worte.
3. Freigegebene Zeit für Einheiten einfacher Länge der Typen %R, %AI und %AQ.
4. Die COMMREQ-Zeit wurde gemessen zwischen CPU und HSC.
5. DOIO ist die Zeit für die Ausgabe von Werten zu diskreten Ausgangsmodulen.
6. Gibt es mehr als einen möglichen Fall, gibt die in der Tabelle angegebene Zeit den ungünstigsten Fall an.
7. Bei Funktionen mit Inkrementwert wird das Inkrement mit (Länge –1) multipliziert. Dieser Wert wird dann zur Grundzeit addiert.

Tabelle A-3. Zeitbedarf SER-Funktionsblock

Konfiguration	Beispiel	Zeit (µs)
Kein Stromfluss (inaktiv)	—	26,50
Zusammenhängend		
8 Kanäle	%I1—8	79,94
16 Kanäle	%I1—16	80,58
24 Kanäle	%I1—24	81,56
32 Kanäle	%I1—32	81,73
8 + 8 zusammenhängende Kanäle	%I1—8 und %Q1—8	111,03
8 + 8 + 8 zusammenhängende Kanäle	%I1—8, %Q1—8 und %M1—8	143,38
8 + 8 + 8 + 8 zusammenhängende Kanäle	%I1—8, %Q1—8 und %M1—8 und %T1—8	175,79
Nicht zusammenhängend	%I1, %M10, %Q3, usw.	
8 Kanäle		299,64
16 Kanäle		552,83
24 Kanäle		806,35
32 Kanäle		1059,85
Rücksetzen		
mit 8 Kanälen	—	162,63
mit 16 Kanälen	—	267,51
mit 24 Kanälen	—	372,73
mit 32 Kanälen	—	477,95

Hinweise:

Geben Sie bei der Angabe eines Eingangsmoduls zu den **zusammenhängenden** und **nicht zusammenhängendem** Zeitwerten jeweils 46 µs hinzu.

Geben Sie bei Auftreten des Triggers weitere 29 µs bei BCD-Format bzw. 148 µs bei POSIX-Format hinzu.

Die für das Rücksetzen angegebenen Zeiten gelten für eine maximale Puffergröße von 1024 Abtastwerten (Rücksetzen löscht alle Abtastwerte im Abtastwertpuffer).

Befehlslängen bei Hochleistungs-CPUs

Die Speichergröße gibt an, wieviel Byte von der Anweisung in einem Kontaktplan-Anwenderprogramm belegt werden. Bei den CPUs 351 und 352 benötigen die meisten Boolesche Standardfunktionen drei Byte — siehe Tabelle A-3.

Tabelle A-4. Befehlslängen für CPUs 350—352, 360, 363 und 364

Funktion	Größe
Keine Operation	1
Nimm von Stack und UNDe an Anfang	1
Nimm von Stack und ODERe an Anfang	1
Kopiere Stackanfang	1
Nimm von Stack	1
Initialisiere Stack	1
Marke	5
Sprung	5
Alle anderen Anweisungen	3
Funktionsblöcke — siehe Tabelle A-2	–

Bearbeitungszeiten Boolescher Funktionen

Die nachstehende Tabelle enthält die Ausführungszeiten von Spulen und Kontakten bei den CPU-Modulen der Series 90-30.

Tabelle A-5. Bearbeitungszeiten Boolescher Funktionen

CPU-Modell	Ausführungszeit pro 1.000 Boolescher Kontakte/Spulen
Modell 350 und 360	0,22 Millisekunden
Modell 340/341	0,3 Millisekunden
Modell 331	0,4 Millisekunden
Modell 313/323	0,6 Millisekunden
Modell 311	18,0 Millisekunden

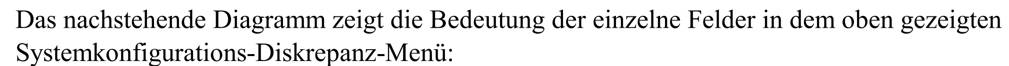
Anhang B Fehlertabellen interpretieren

Die speicherprogrammierbaren Steuerungen der Series 90-30, 90-20 und Micro verwalten zwei Fehlertabellen, die E/A-Fehlertabelle für Fehler, die von E/A-Geräten erzeugt wurden (einschließlich E/A-Controller) und die SPS-Fehlertabelle für interne SPS-Fehler. Die Informationen in diesem Anhang sollen Ihnen helfen, das Format der Meldungsstruktur beim Lesen dieser Fehlertabellen zu interpretieren. In beiden Tabellen finden Sie ähnliche Informationen.

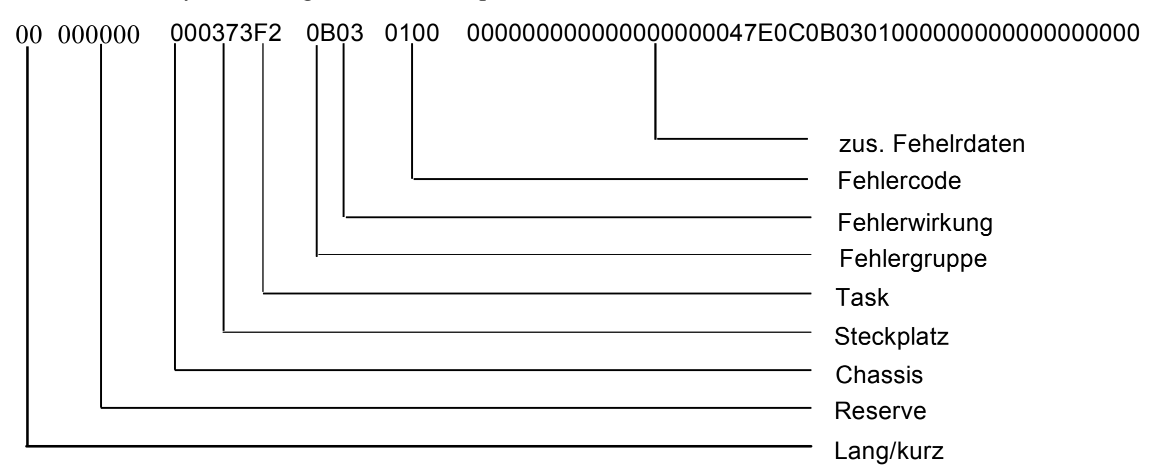
- Die SPS-Fehlertabelle enthält:
  - Fehleradresse.
  - Fehlerbeschreibung.
  - Datum und Zeit des Fehlers.
- Die E/A-Fehlertabelle enthält:
  - Fehleradresse.
  - Referenzadresse.
  - Fehlerkategorie.
  - Fehlertyp.
  - Datum und Zeit des Fehlers.

SPS-Fehlertabelle

Rufen Sie die SPS-Fehlertabelle über Ihre Programmiersoftware auf. Informationen zum Zugriff auf Fehlertabellen finden Sie in der Online-Hilfe, *Logicmaster 90 Series 90-30/20/Micro Programmiersoftware Anwenderhandbuch*, GFK-0466.



Das nachstehende Diagramm zeigt die Bedeutung der einzelne Felder in dem oben gezeigten Systemkonfigurations-Diskrepanz-Menü:



Der Fehlereintrag "Diskrepanz bei Systemkonfiguration" wird nachstehend erläutert (alle Daten in Hexadezimaldarstellung) (alle Daten in Hexadezimaldarstellung).

Feld	Wert	Beschreibung
Lang/kurz	00	Dieser Fehler enthält 8 Byte zusätzliche Fehlerdaten.
Chassis	00	Hauptchassis (Chassis 0)
Steckplatz	03	Steckplatz 3
Task	44	
Fehlergruppe	0B	Diskrepanz bei Systemkonfiguration
Fehlerwirkung	03	Fataler Fehler
Fehlercode	01	

In den folgenden Abschnitten werden die einzelnen Felder des Fehlereintrags erläutert. Außerdem wird tabellarisch angegeben, welche Werte die einzelnen Felder annehmen können.

Anzeige Lang/kurz

Dieses Byte gibt an, ob der Fehler 8 oder 24 Byte zusätzlicher Fehlerdaten enthält.

Typ	Code	Zusätzliche Fehlerdaten
Kurz	00	8 Byte
Lang	01	24 Byte

Reserve

Mit diesen sechs Füllbyte wird der SPS-Fehlertabelleneintrag genauso lang wie der E/A-Fehlertabelleneintrag.

Chassis

Die Chassisnummer liegt zwischen 0 und 7. Null ist das Hauptchassis mit der CPU. Die Chassis 1 bis 7 sind Erweiterungschassis, die mit dem CPU-Chassis über ein Erweiterungskabel verbunden sind.

Steckplatz

Die Steckplatznummer liegt zwischen 0 und 9. Die CPU der SPS belegt immer Steckplatz 1 im Hauptchassis (Chassis 0).

Task

Die Tasknummer liegt zwischen 0 und +65,535. Manchmal liefert die Tasknummer zusätzliche Informationen für SPS-Ingenieure. Normalerweise kann die Tasknummer aber ignoriert werden.

SPS-Fehlergruppe

Fehlergruppe ist die oberste Klassifizierungsebene eines Fehlers. Sie gibt die allgemeine Fehlerkategorie an. Der von der Logicmaster 90-30/20/Micro Software angezeigte Text basiert auf Fehlergruppe und Fehlercodes.

Tabelle B-1 listet die in der SPS-Fehlertabelle möglichen Fehlergruppen auf.

Die letzte nicht maskierbare Fehlergruppe, ***weitere SPS-Fehlercodes***, gilt für die Bearbeitung neuer Fehlerzustände im System, ohne dass die SPS die Alarmcodes besonders kennen muss. Sämtliche nicht identifizierten SPS-Alarmcodes gehören zu dieser Gruppe.

Tabelle B-1. SPS-Fehlergruppen

Gruppen-nummer			
Dezimal	**Hexadezimal**	**Gruppenname**	**Fehlerwirkung**
1	1	Verlorenes oder fehlendes Chassis	Fatal
4	4	Verlorenes oder fehlendes Zusatzmodul	Diagnose
5	5	Hinzugefügtes oder überzähliges Chassis	Diagnose
8	8	Hinzugefügtes oder überzähliges Zusatzmodul	Diagnose
11	B	Diskrepanz bei Systemkonfiguration	Fatal
12	C	Systembusfehler	Diagnose
13	D	SPS CPU Hardwarefehler	Fatal
14	E	Nicht-fataler Modulhardwarefehler	Diagnose
16	10	Softwarefehler bei Zusatzmodul	Diagnose
17	11	Programmblock-Prüfsummenfehler	Fatal
18	12	Signal "niedrige Batteriespannung"	Diagnose
19	13	Konstante Zykluszeit überschritten	Diagnose
20	14	SPS-Systemfehlertabelle ist voll	Diagnose
21	15	E/A-Fehlertabelle ist voll	Diagnose
22	16	Anwenderfehler	Diagnose
–	–	Weitere SPS-Fehlercodes	Wie angegeben
128	80	Systembusfehler	Fatal
129	81	Kein Anwenderprogramm beim Einschalten	Informatorisch
130	82	Verstümmeltes Anwender-RAM erkannt	Fatal
132	84	Passwortfehler	Informatorisch
135	87	SPS CPU Softwarefehler	Fatal
137	89	Fehler beim Speichern von SPS-Sequenzen	Fatal

Fehlerwirkung

Mit jedem Fehler kann eine von drei Fehlerwirkungen verknüpft sein. Bei der SPS Series 90-30 sind die Fehlerwirkungen unveränderlich festgelegt.

Tabelle B-2. SPS-Fehlerwirkungen

Fehlerwirkung	von CPU durchgeführte Aktion	Code
Informatorisch	Fehlereintrag in Fehlertabelle	1
Diagnose	Fehlereintrag in Fehlertabelle Setzen von Fehlerreferenzen	2
Fatal	Fehlereintrag in Fehlertabelle Setzen von Fehlerreferenzen Übergang in STOP-Modus	3

Fehlercode

Der Fehlercode liefert eine weitere Beschreibung des Fehlers. Jede Fehlergruppe besitzt ihren eigenen Satz Fehlercodes. Tabelle B-3 zeigt die Fehlercodes für Gruppe 87H (SPS-Softwarefehler).

Tabelle B-3. Alarm-Fehlercodes für SPS CPU-Softwarefehler

Dezimal	Hexadezimal	Bezeichnung
20	14	Verstümmelter SPS-Programmspeicher
39	27	Verstümmelter SPS-Programmspeicher
82	52	Fehler bei Kommunikation über Rückwand-platine
90	5A	Anforderung anwenderseitige Abschaltung
Alle übrigen.		Interner Systemfehler der SPS CPU

Tabelle B-4 zeigt die Fehlercodes für alle anderen Fehlergruppen.

Tabelle B-4. Alarm-Fehlercodes für SPS-Fehler

Dezimal	Hexadezimal	Bezeichnung
SPS-Fehlercodes für Gruppe "Verlust von Zusatzmodul" (4)		
44	2C	Software-Rücksetzen von Zusatzmodul gescheitert
45	2D	Software-Rücksetzen von Zusatzmodul gescheitert
255	FF	Kommunikation mit Zusatzmodul fehlerhaft
79	4F	Verlust des Daughterboards
Fehlercodes für Gruppe "Rücksetzen von, hinzugefügtes oder überzähliges Zusatzmodul" (8)		
2	2	Modul-Neustart abgeschlossen
04	4	Hinzugefügtes Daughterboards
05	5	Rücksetzen des Daughterboards
	Alle übrigen.	Rückgesetztes, hinzugefügtes oder überzähliges Zusatzmodul
Fehlercodes für Gruppe "Zusatzmodul-Softwarefehler" (10 hexa)		
1	1	Modultyp nicht unterstützt
2	2	COMREQ-Mailbox voll bei ausgehender Meldung, die COMREQ startet.
3	3	COMREQ-Mailbox voll bei Antwort
5	5	Rückwandplatinenkommunikation mit SPS; verlorene Anforderung
11	B	Betriebsmittelfehler (alloc, tbl ovrflw, usw.)
13	D	Anwenderprogrammfehler
401	191	Modulsoftware fehlerhaft; Neuladen gefordert
Fehlercodes für Gruppe "Diskrepanz bei Systemkonfiguration" (B hexa)		
8	8	Diskrepanz bei Analogerweiterung
10	A	Nicht unterstützte Funktion
23	17	Programm überschreitet Speichergrenzen
58	3 A	Diskrepanz bei Daughterboard
Fehlercodes für Gruppe "Systembusfehler" (C hexa)		
Alle übrigen.		Systembusfehler
Fehlercodes für Gruppe "Programmblock-Prüfsummenfehler" (11 hexa)		
3	3	Prüfsummenfehler bei Programm oder Programmblock
Fehlercodes für Signal "niedrige Batteriespannung"		
0	0	Batterie in CPU- oder anderem Modul ausgefallen
1	1	Batteriespannung in CPU- oder anderem Modul niedrig
Fehlercodes für Gruppe "Anwenderprogrammfehler" (16 hexa)		
2	2	SPS-Zeitüberwachung (Watchdog) abgelaufen
5	5	COMREQ WAIT-Modus für diesen Befehl nicht verfügbar
6	6	COMREQ – falsche Task-ID
7	7	Überlauf Anwenderstack
Fehlercodes für Gruppe "Systembusfehler" (80 hexa)		
1	1	Betriebssystem
Fehlercodes für Gruppe "Verstümmeltes Anwender-RAM beim Einschalten" (82 hexa)		
1	1	Verstümmeltes Anwender-RAM beim Einschalten
2	2	Unzulässiger Boolescher Operationscode erkannt
3	3	PLC_ISCP_PC_OVERFLOW
4	4	PRG_SYNTAX_ERR
Fehlercodes für SPS-CPU-Hardwarefehler (D hexa)		
Alle Codes		SPS CPU Hardwarefehler

Zusätzliche Fehlerdaten

Dieses Feld enthält Einzelheiten zum Fehlereintrag. Das folgende Beispiel zeigt, welche Daten vorhanden sein können:

Gruppe defekter Anwender-RAM:

Vier der Fehlercodes in der Gruppe "Diskrepanz bei Systemkonfiguration" liefern zusätzliche Fehlerdaten:

Tabelle B-5. SPS-Fehlerdaten - Unzulässiger Boolescher Operationscode erkannt

Zusätzliche Fehlerdaten	Modellnummer-Diskrepanz
[0]	Inhalt ISCP-Fehlerregister
[1]	Fehlerhafter OPCODE
[2,3]	ISCP-Programmzähler
[4,5]	Funktionsnummer

Bei einem RAM-Fehler in der SPS-CPU (einer der als SPS-CPU-Hardwarefehler gemeldeten Fehler) wird die Adresse des Fehlers in den ersten vier Byte des Feldes gespeichert.

SPS CPU Hardware-Fehler (RAM-Fehler):

SPS-Fehlerzeitstempel

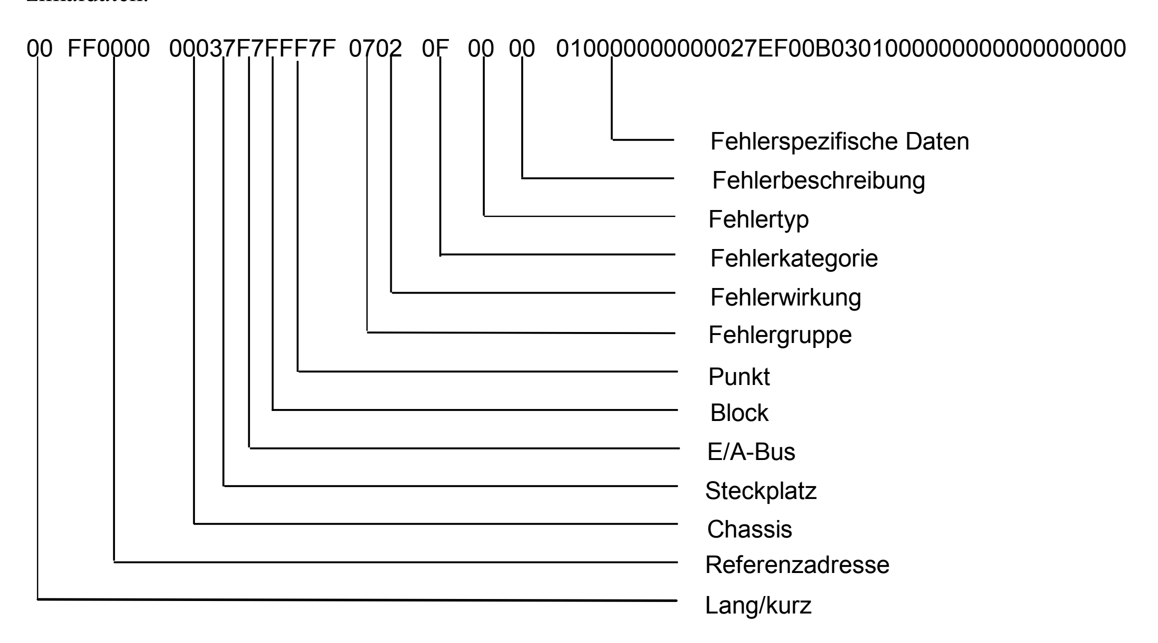
Der sechs Byte lange Zeitstempel gibt den Wert an, den die Systemuhr hatte, als der Fehler von der SPS-CPU gemeldet wurde (die Werte sind im BCD-Format).

Tabelle B-6. SPS-Fehlerzeitstempel

Bytenummer	Beschreibung
1	Sekunden
2	Minuten
3	Stunden
4	Tag
5	Monat
6	Jahr

E/A-Fehlertabelle

Das folgende Diagramm zeigt die in den einzelnen Feldern des Fehlereintrags enthaltenen Hexadezimaldaten.



In den folgenden Abschnitten werden die einzelnen Felder der E/A-Fehlertabelle erläutert. Außerdem wird tabellarisch angegeben, welche Werte die einzelnen Felder annehmen können.

Anzeige Lang/kurz

Dieses Byte gibt an, ob der Fehler 5 oder 21 Byte fehlerspezifische Daten enthält.

Tabelle B-7. E/A-Fehlertabelle, Formatanzeigebyte

Typ	Code	Fehlerspezifische Daten
Kurz	02	5 Byte
Lang	03	21 Byte

Referenzadresse

Die Referenzadresse besteht aus drei Byte und enthält den E/A-Speichertyp und die Speicheradresse (oder Offset) in diesem Speicher, die dem fehlerbehafteten Punkt entspricht. Tritt ein Fehler bei einem Genius-Block oder einem Analogmodul auf, zeigt die Referenzadresse auf den ersten Punkt des Blocks, in dem der Fehler aufgetreten ist.

Tabelle B-8. E/A-Referenzadresse

Byte	Beschreibung	Bereich
0	Speichertyp	0 – FF
1 – 2	Offset	0 – 7FF

Das Speichertypbyte kann einen der folgenden Werte annehmen:

Tabelle B-9. Speichertyp der E/A-Referenzadresse

Name	Wert (hexadezimal)
Analogeingang	0A
Analogausgang	0C
Analoggruppe	0D
Diskreter Eingang	10 oder 46
Diskreter Ausgang	12 oder 48
Diskrete Gruppe	1F

E/A-Fehleradresse

Die E/A-Fehleradresse ist sechs Byte lang und gibt an, bei welchem Chassis, Steckplatz, Bus, Block und Punkt der Fehler aufgetreten ist. Die Punktadresse ist ein Wort, alle anderen Adressen jeweils ein Byte. Ein Fehler muss nicht alle fünf Werte enthalten.

Enthält eine E/A-Fehleradresse nicht alle fünf Adresswerte, zeigt ein Hexadezimalwert von 7F an, wo die Signifikanz aufhört. Steht zum Beispiel 7F im Busbyte, handelt es sich um einen Modulfehler, bei dem nur die Werte für Chassis und Steckplatz signifikant sind.

Chassis

Die Chassisnummer liegt zwischen 0 und 7. Null ist das Hauptchassis mit der CPU. Die Chassis 1 bis 7 sind Erweiterungschassis.

Steckplatz

Die Steckplatznummer liegt zwischen 0 und 9. Die CPU der SPS belegt immer Steckplatz 1 im Hauptchassis (Chassis 0).

Punkt

Die Punktnummer liegt zwischen 1 und 1024 (dezimal). Sie gibt bei einem Punktfehler an, welcher Punkt des Blockes fehlerbehaftet ist.

E/A-Fehlergruppe

Fehlergruppe ist die oberste Klassifizierungsebene eines Fehlers. Sie gibt die allgemeine Fehlerkategorie an. Der von der Logicmaster 90-30/20/Micro Software angezeigte Text basiert auf Fehlergruppe und Fehlercodes.

Tabelle B-10 listet die in der E/A-Fehlertabelle möglichen Fehlergruppen auf. Bei Gruppennummern kleiner als 80 (Hexa) handelt es sich um maskierbare Fehler.

Die letzte nicht maskierbare Fehlergruppe, ***weitere E/A-Fehlercodes***, gilt für die Bearbeitung neuer Fehlerzustände im System, ohne dass die SPS die Alarmcodes besonders kennen muss. Sämtliche nicht identifizierten E/A-Alarmcodes gehören zu dieser Gruppe.

Tabelle B-10. E/A-Fehlergruppen

Gruppennummer	Gruppenname	Fehlerwirkung
3	Verlorenes oder fehlendes E/A-Modul	Diagnose
7	Hinzugefügtes oder überzähliges E/A-Modul	Diagnose
9	IOC- oder I/O-Bus-Fehler	Diagnose
A	E/A-Modulfehler	Diagnose
–	Weitere E/A-Fehlercodes	Wie angegeben

E/A-Fehlerwirkung

Die Fehlerwirkung gibt an, welche Maßnahmen die CPU der SPS beim Auftreten eines Fehlers ergreifen soll. Tabelle B-11 gibt die möglichen Fehlerwirkungen an.

Tabelle B-11. E/A-Fehlerwirkungen

Fehlerwirkung	von CPU durchgeführte Aktion	Code
Informatorisch	Fehlereintrag in Fehlertabelle	1
Diagnose	Fehlereintrag in Fehlertabelle Setzen von Fehlerreferenzen	2
Fatal	Fehlereintrag in Fehlertabelle Setzen von Fehlerreferenzen Übergang in STOP-Modus	3

E/A-Fehlerdaten

Ein Eintrag in der E/A-Fehlertabelle kann bis zu 5 Byte fehlerspezifische Daten enthalten.

Symbolische fehlerspezifische Daten

Tabelle B-12 gibt die für die Konfiguration der Blockschaltkreise benötigten Daten an.

Tabelle B-12. E/A-Fehlerspezifische Daten

Dezimalzahl	Hexa-Code	Beschreibung
	Schaltkreiskonfiguration	
	1 2 3	Anschluss ist ein Eingang – Tristate Anschluss ist ein Eingang Anschluss ist ein Ausgang

Fehlerwirkungen für bestimmte Fehler

Fixierte/unfixierte Schaltkreisfehler werden als informatorische Fehler gemeldet. Alle übrigen sind diagnostisch oder fatal.

Diskrepanz bei Modellnummer oder E/A-Typ sowie nicht vorhandene E/A-Module werden in der SPS-Fehlertabelle unter der Gruppe "Diskrepanz bei Systemkonfiguration" gemeldet. Sie werden nicht in der E/A-Fehlertabelle gemeldet.

E/A-Zeitstempel

Der sechs Byte lange Zeitstempel gibt den Wert an, den die Systemuhr hatte, als der Fehler von der SPS-CPU gemeldet wurde (die Werte sind im BCD-Format).

Tabelle B-13. E/A-Zeitstempel

Bytenummer	Beschreibung
1	Sekunden
2	Minuten
3	Stunden
4	Tag
5	Monat
6	Jahr

Anhang C Mnemonische Programmieranweisungen

Im Programmanzeige-/Editiermodus können Sie auf eine schnelle Art Programmieranweisungen eingeben oder suchen, indem Sie ein "&"-Zeichen und die mnemonische Bezeichnung der Anweisung eingeben. Bei einigen Anweisungen können Sie auch eine Referenzadresse oder eine symbolische Adresse, einen Kennsatz (Label) oder eine Speicher-Referenzadresse eingeben.

Die Tabellen in diesem Anhang enthalten die mnemonischen Bezeichnungen der Programmanweisungen der Logicmaster 90-30/20/Micro Programmiersoftware. Die komplette Mnemonik steht in Spalte 3 der Tabelle, die kürzestmögliche Eingabe in Spalte 4.

Während der Programmierung können Sie jederzeit ein Hilfsmenü zu diesen mnemonischen Bezeichnungen aufrufen, indem Sie die Tastenkombination ALT-I drücken.

Funktions-gruppe	**Anweisung**	**Mnemonik**						
		Alle	INT	DINT	BIT	BYTE	WORD	REAL
Kontakte	Beliebiger Kontakt	&CON	&CON					
	Schließerkontakt (NO)	&NOCON	&NOCON					
	Öffnerkontakt (NC)	&NCCON	&NCCON					
	Fortsetzkontakt	&CONC	&CONC					
Spulen	Beliebige Spule	&COI	&COI					
	Schließer	&NOCOI	&NOCOI					
	Negierte Spule	&NCCOI	&NCCOI					
	Spule für positive Übergänge	&PCOI	&PCOI					
	Spule für negative Übergänge	&NCOI	&NCOI					
	SET-Spule	&SL	&SL					
	RESET-Spule	&RL	&RL					
	Remanente SET-Spule	&SM	&SM					
	Remanente RESET-Spule	&RM	&RM					
	Remanente Spule	&NOM	&NOM					
	Negierte remanente Spule	&NCM	&NCM					
	Fortsetzspule	&COILC	&COILC					
Verbindungen	Horizontalverbindung	&HO	&HO					
	Vertikalverbindung	&VE	&VE					
Timer	Einschaltverzögerung	&ON	&ON					
	Zeiterfassung	&TM	&TM					
	Abschaltverzögerung	&OF	&OF					
Zähler	Aufwärtszähler	&UP	&UP					
	Abwärtszähler	&DN	&DN					

Funktions-gruppe	Anweisung	Mnemonik							
		Alle	BCD-4	INT	DINT	BIT	BYTE	WORD	REAL
Arithmetik-funktionen	Addition	&AD		&AD_I	&AD_DI				&AD_R
	Subtraktion	&SUB		&SUB_I	&SUB_DI				&SUB_R
	Multiplikation	&MUL		&MUL_I	&MUL_DI				&MUL_R
	Division	&DIV		&DIV_I	&DIV_DI				&DIV_R
	Modulo	&MOD		&MOD_I	&MOD_DI				&MOD_R&SQ_R
	Quadratwurzel	&SQ		&SQ_I	&SQ_DI				
	Sinus	&SIN							
	Kosinus	&COS							
	Tangens	&TAN							
	Arkussinus	&ASIN							
	Arkuskosinus	&ACOS							
	Arkustangens	&ATAN							
	Dekadischer Logarithmus	&LOG							
	Natürlicher Logarithmus	&LN							
	Natürliche Exponential-funktion	&EXP							
	Potenz von X	&EXPT							
Relationale Funktionen	Gleich	&EQ		&EQ_I	&EQ_DI				&EQ_R
	Ungleich	&NE		&NE_I	&NE_DI				&NE_R
	Größer als	&GT		&GT_I	&GT_DI				&GT_R
	Größer als oder gleich	&GE		&GE_I	&GE_DI				&GE_R
	Kleiner als	&LT		&LT_I	&LT_DI				&LT_R
	Kleiner als oder gleich	&LE		&LE_I	&LE_DI				&LE_R
Bit-operationen	AND	&AN						&AN_W	
	OR	&OR						&OR_W	
	Antivalenz	&XO						&XO_W	
	NOT	&NOT						&NOT_W	
	Bit nach links schieben	&SHL						&SHL_W	
	Bit nach rechts schieben	&SHR						&SHR_W	
	Bit nach links rotieren	&ROL						&ROL_W	
	Bit nach rechts rotieren	&ROR						&ROR_W	
	Bit testen	&BT						&BT_W	
	Bit auf 1 setzen	&BS						&BS_W	
	Bit auf 0 setzen	&BCL						&BCL_W	
	Bitposition	&BP						&BP_W	
	Vergleich mit Maske	&MCMP						&MCM_W	
Umwandlung	Umwandlung in ganze Zahl	&TO_INT	&TO_INT_BCD4	&MOV					
	Umwandlung in doppelt-genaue ganze Zahl	&TO_DINT		&BLKM					&BCD4_R
	Umwandlung in BCD–4	&BCD4		&BLKC	&TO_REAL_DI				
	Umwandlung in REAL	&TO_REAL		&SHF				&TO_REAL_W	
	Umwandlung in WORD	&TO_W		&BI					
	Kürze auf ganze Zahl	&TRINT		&COMMR					
	Kürze auf doppeltgenaue ganze Zahl	&TRDINT							

Funktionsgruppe	Anweisung	Mnemonik						
		Alle	**INT**	**DINT**	**BIT**	**BYTE**	**WORD**	**REAL**
Daten kopieren	Kopieren	&MOV	&MOV_I		&MOV_BI		&MOV_W	&MOV_R
	Block kopieren	&BLKM	&BLKM_I				&BLKM_W	&BLKM_R
	Block löschen	&BLKC						
	Schieberegister	&SHF			&SHF_BI		&AR_W	
	Bitfolgesteuerung	&BI						
	Kommunikationsanforderung	&COMMR						
Tabelle	Feld kopieren	&AR	&AR_I	&AR_DI	&AR_BI	&AR_BY	&AR_W	
	Suche gleiche	&SRCHE	&SRCHE_I	&SRCHE_DI		&SRCHE_BY	&SRCHE_W	
	Suche ungleiche	&SRCHN	&SRCHN_I	&SRCHN_DI		&SRCHN_BY	&SRCHN_W	
	Suche größer als	&SRCHGT	&SRCHGT_I	&SRCHGT_DI		&SRCHGT_BY	&SRCHGT_W	
	Suche größer als/gleich	&SRCHGE	&SRCHGE_I	&SRCHGE_DI		&SRCHGE_BY	&SRCHGE_W	
	Suche kleiner als	&SRCHLT	&SRCHLT_I	&SRCHLT_DI		&SRCHLT_BY	&SRCHLT_W	
	Suche kleiner als/gleich	&SRCHLE	&SRCHLE_I	&SRCHLE_DI		&SRCHLE_BY	&SRCHLE_W_	
Steuerung	Unterprogrammaufruf	&CA						
	E/A-Aktualisierung	&DO						
	SER	&SER						
	PID – ISA-Algorithmus	&PIDIS						
	PID – IND-Algorithmus	&PIDIN						
	SFC-Rücksetzen	&SFCR						
	Ende	&END						
	Strompfadkommentar	&COMME						
	System-Bedienanforderung	&SV						
	Hauptsteuerrelais	&MCR						
	Hauptsteuerrelais-Ende	&ENDMCR						
	Geschachteltes Hauptsteuerrelais	&MCRN						
	Ende geschachteltes Hauptsteuerrelais	&ENDMCRN						
	Sprung	&JUMP						
	Geschachtelter Sprung	&JUMPN						
	Marke	&LABEL						
	Geschachtelte Marke	&LABELN						

Anhang D Tastenfunktionen

In diesem Anhang werden die in der Softwareumgebung aktiven Tastaturfunktionen beschrieben. Diese Informationen können Sie auch mit der Tastenkombination ALT-K auf dem Bildschirm des Programmiergerätes anzeigen.

Tastenfolge	Beschreibung	Tastenfolge	Beschreibung
Diese Tasten sind im gesamten Softwarepaket verfügbar			
ALT-A	Abbruch	CTRL-Break	Logicmaster 90 verlassen.
ALT-C	Feld löschen.	Esc	Zu höherer Funktionsebene umschalten.
ALT-M	Programmiergerätemodus umschalten.	CTRL-Home	Vorherige Befehlszeile anzeigen.
ALT-R	SPS-Modus (RUN/STOP) umschalten.	CTRL-End	Nächste Befehlszeile anzeigen.
ALT-E	Statusbereich umschalten.	CTRL- ←	Cursor nach links im Feld bewegen.
ALT-J	Befehlszeile umschalten.	CTRL-→	Cursor nach rechts im Feld bewegen.
ALT-L	Plattendirectory anzeigen.	CTRL-D	Referenzadresse dekrementieren.
ALT-P	Bildschirminhalt drucken.	CTRL-U	Referenzadresse inkrementieren.
ALT-H	Hilfe.	Tab	Eingabefeld ändern/inkrementieren.
ALT-K	Liste der Tastaturfunktionen anzeigen.	Shift-Tab	Eingabefeld ändern/dekrementieren.
ALT-I	Hilfe, mnemonische Anweisungen.	Eingabetaste	Feldinhalt übernehmen.
ALT-N	Anzeigeoptionen umschalten.	CTRL-E	Anzeige des letzten Systemfehlers.
ALT-T	Lernmodus Start.	F12 oder Zehner-tastatur -	Diskrete Referenz umschalten.
ALT-Q	Lernmodus Stop.	F12 oder Zehner-tastatur *	Diskrete Referenz überschreiben.
ALT-N	Playback-Datei n (n = 0 bis 9).		
Diese Tasten sind nur im Programmeditor verfügbar			
ALT-B	Glocke beim Texteditieren ein/aus.	Zehnertastatur +	Strompfad übernehmen.
ALT-D	Strompfadelement/Strompfad löschen.	Eingabetaste	Strompfad übernehmen.
ALT-S	Block in SPS und auf Platte speichern.	CTRL-PgUp	Vorherigen Strompfad anzeigen.
ALT-X	Anzeige Zoomebene.	CTRL-PgDn	Nächsten Strompfad anzeigen.
ALT-U	Platte aktualisieren.	~	Horizontalverbindung.
ALT-V	Variablentabellen-Fenster.	|	Vertikalverbindung.
ALT-F2	Gehe zu Operandenreferenztabelle.	Tab	Gehe zum nächsten Operandenfeld.
Spezialtasten			
ALT-O	Passwort überschreiben. Nur im Passwortmenü der Konfigurationssoftware verfügbar.		

Die Hilfekarte auf der nächsten Seite enthält eine Liste der Tastenfunktionen und der mnemonischen Programmieranweisungen für die Logicmaster 90-30/20/Micro Software. Diese Karte ist dreifach vorhanden und kann durch ihre Perforation leicht aus dem Handbuch heraus getrennt werden.

Seite 1 von GFJ-055D auf diese Seite drucken.

Seite 2 von GFJ-055D auf diese Seite drucken.

Anhang E Verwendung von Fliesskommazahlen

Bei der Verwendung von Fliesskommazahlen gibt es einige Punkte zu beachten. Im ersten Abschnitt werden die allgemeinen Überlegungen behandelt. Anweisungen zur Eingabe und Anzeige von Fliesskommazahlen finden Sie ab Seite E-5.

Hinweis

Fliesskommafunktionen werden ***nur*** bei CPUs der Modelle 35x und 36x, Ausgabestand 9 oder höher und bei allen Ausgabeständen der CPU352 unterstützt.

Fliesskommazahlen

Mit der Programmiersoftware können Sie Zahlen mit reellen Werten bearbeiten, anzeigen, speichern und aufrufen. Einige Funktionen verarbeiten Fliesskommazahlen. Sie können jedoch Fliesskommazahlen mit der Logicmaster 90-30/20/Micro Software nur im Zusammenhang mit einer CPU 35x oder 36x verwenden. Fliesskommazahlen werden in wissenschaftlicher Dezimalschreibweise dargestellt, wobei sechs signifikante Stellen angezeigt werden.

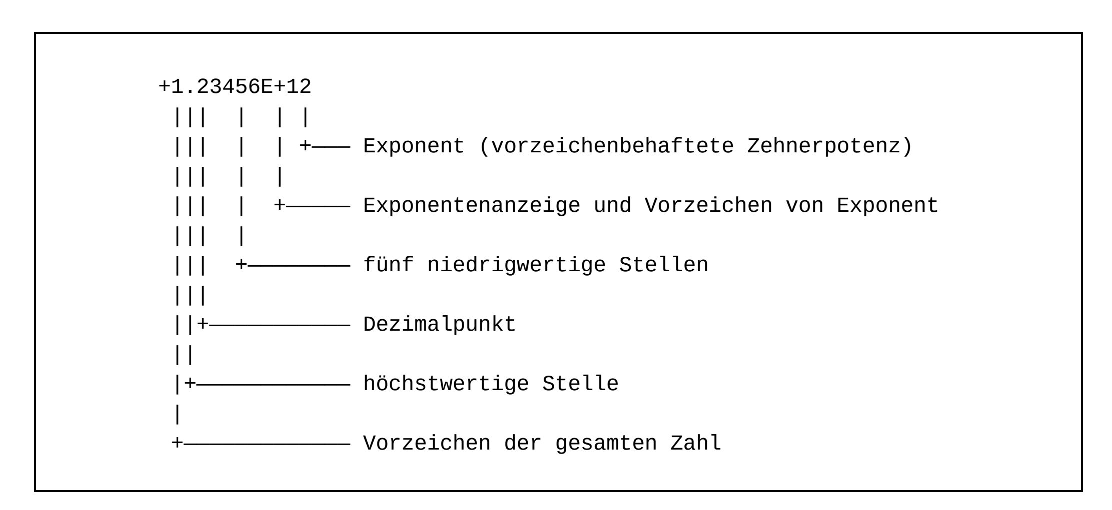
Hinweis

In diesem Handbuch werden die Begriffe "Fliesskomma" und "REAL" gleichberechtigt zur Beschreibung der Fliesskomma-Anzeige- bzw. Eingabefunktion der Programmiersoftware verwendet.

Es wird folgendes Format benutzt. Bei Zahlen im Bereich zwischen 99999999 und 0,0001 besitzt die Anzeige keinen Exponenten und bis zu sechs oder sieben signifikante Stellen. Zum Beispiel:

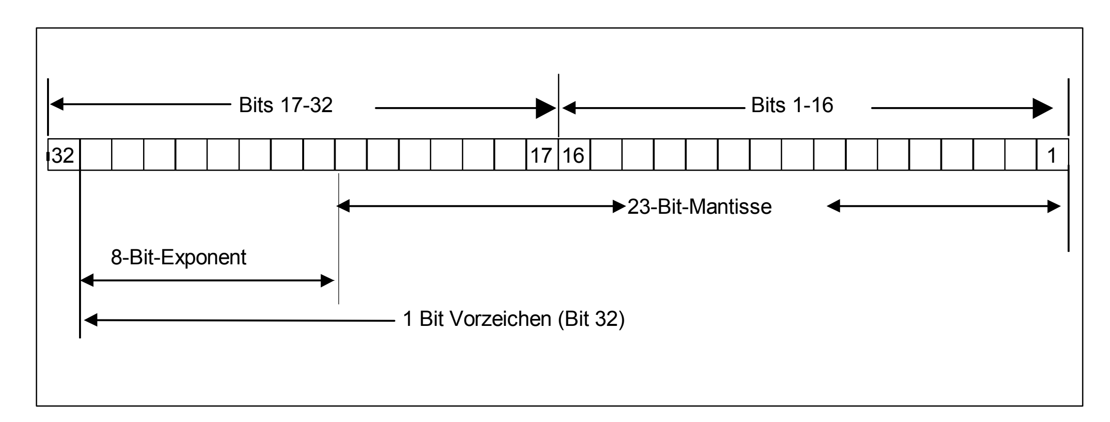
Eingabe	Anzeige	Beschreibung
.000123456789	+.0001234567	Zehn Stellen, sechs oder sieben signifikant.
–12.345e-2	–.1234500	Sieben Stellen, sechs oder sieben signifikant.
1234	+1234.000	Sieben Stellen, sechs oder sieben signifikant.

Außerhalb des oben aufgeführten Bereichs werden nur sechs Stellen in der folgenden Form angezeigt:

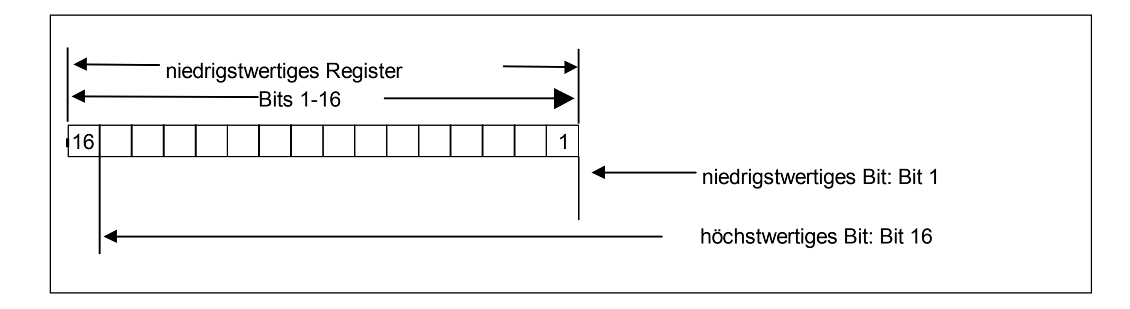


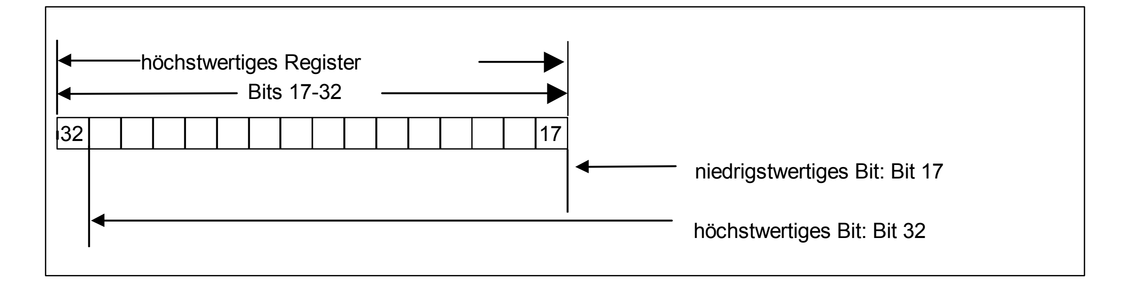
Internes Format der Fliesskommazahlen

Fliesskommazahlen werden im einfach genauen IEEE-Standardformat abgelegt. Dieses Format benötigt 32 Bits, die in zwei nebeneinanderliegenden 16-Bit-SPS-Registern aufgeteilt werden. Die nachstehenden Zeichnungen zeigen die Kodierung der Bits.



Die nachstehenden Zeichnungen zeigen die Registerbelegung durch eine einzelne Fliesskommazahl. Belegt die Fliesskommazahl zum Beispiel die Register R5 und R6, ist in diesem Diagramm R5 das niedrigstwertige Register und R6 das höchstwertige Register.





Werte der Fliesskommazahlen

Mit der folgenden Tabelle können Sie den Wert einer Fliesskommazahl aus der in zwei Registern gespeicherten Binärzahl berechnen.

Exponent (e)	Mantisse (f)	Wert der Fliesskommazahl
255	ungleich 0	keine gültige Zahl
255	0	$-1^{s} * \infty$
0 < e < 255	beliebiger Wert	$-1^{s} * 2^{e-127} * 1.f$
0	ungleich 0	$-1^{s} * 2^{-126} * 0.f$
0	0	0

f = Die Mantisse. Die Mantisse ist ein binärer Bruch.
e = Der Exponent. Der Exponent ist eine ganze Zahl E, so dass E+127 die Zweierpotenz ist, mit der die Mantisse multipliziert werden muss, um den Gleitpunktwert zu erhalten.
s = Das Vorzeichenbit.
* = Der Multiplikationsoperator.

Betrachten wir zum Beispiel die Fliesskommazahl 12.5. Die binäre IEEE-Fliesskommadarstellung dieser Zahl ist:

```
01000001 01001000 00000000 00000000
```

oder, im Hexadezimalformat, 41480000H. Das höchstwertige Bit (das Vorzeichenbit) ist Null (s=0). Die nächsten acht höchstwertigen Bits sind 10000010, oder dezimal 130 (e=130).

Die Mantisse wird als binäre Dezimalzahl gespeichert, bei der der Dezimalpunkt vor dem höchstwertigen der 23 Bits steht. Das höchstwertige Bit in der Mantisse ist daher ein Vielfaches von 2^{-1}, das nächste Bit in der Wertigkeit ein Vielfaches von 2^{-2}, usw. bis zum niedrigstwertigen Bit, das ein Vielfaches von 2^{-23} ist. Die letzten 23 Bits (die Mantisse) sind:

```
1001000 00000000 00000000
```

Der Wert der Mantisse ist dann .5625 (d.h. $2^{-1} + 2^{-4}$).

Da e > 0 und e < 255, benutzen wir die dritte Formel in der obenstehenden Tabelle:

$$
\begin{aligned}
\text{Zahl} &= -1^{s} * 2^{e-127} * 1.f \\
&= -1^{0} * 2^{130-127} * 1.5625 \\
&= 1 * 2^{3} * 1.5625 \\
&= 8 * 1.5625 \\
&= 12.5
\end{aligned}
$$

Wie Sie sehen können, ist die obenstehende binäre Darstellung richtig.

In diesem Format können die Zahl Null sowie der Zahlenbereich von ± 1.401298E–45 bis ± 3.402823E+38 gespeichert werden.

Eingabe und Anzeige von Fliesskommazahlen

Obwohl in der Mantisse bis zu sechs oder sieben für die Genauigkeit signifikante Stellen gespeichert werden können, zeigt die Programmiersoftware nur die ersten sechs dieser Stellen an. Vor der Mantisse kann ein positives oder ein negatives Vorzeichen stehen. Wird kein Vorzeichen eingegeben, wird die Mantisse als positiv angenommen.

Wird ein Exponent eingegeben, muss er mit dem Buchstaben **E** oder **e** beginnen und die Mantisse muss einen Dezimalpunkt enthalten, damit sie nicht mit einer Hexadezimalzahl verwechselt wird. Vor dem Exponenten kann ein Vorzeichen stehen. Wird kein Vorzeichen eingegeben, wird der Exponent als positiv angenommen. Wird kein Exponent eingegeben, wird der Exponent als Null angenommen. In einer Fliesskommazahl sind keine Leerzeichen zulässig.

Um die Anwendung einfach zu machen sind bei Einträgen in Befehlszeile und Prozessdaten mehrere Formate möglich. Hierzu gehören ganze Zahlen, Dezimalzahlen, oder Dezimalzahlen mit Exponent. Nachdem der Anwender diese Zahlen eingegeben und die Taste **Enter** gedrückt hat, werden diese Zahlen in eine Standard-Anzeigeform umgewandelt.

In der nachstehenden Tabelle finden Sie Beispiele, wie Fliesskommazahlen eingegeben werden und wie ihre genormte Anzeige aussieht.

Eingabe	Anzeige
250	+250,0000
+4	+4.000000
–2383019	–2383019.
34.	+34.00000
–.0036209	–.003620900
12.E+9	+1.20000E+10
–.0004E–11	–4.00000E–15
731.0388	+731.0388
99.20003e–29	+9.92000E–28

Die nachstehende Tabelle zeigt Beispiele unzulässiger Eingaben von Fliesskommazahlen.

Unzulässige Eingabe	Erläuterung
–433E23	Dezimalpunkt fehlt.
10e-19	Dezimalpunkt fehlt.
10.e19	In der Mantisse sind zwischen den Ziffern oder Zeichen keine Leerstellen zulässig. Diese Eingabe wird als 10.e0 angenommen und es wird eine Fehlermeldung angezeigt.
4.1e19	Im Exponenten sind zwischen den Ziffern oder Zeichen keine Leerstellen zulässig. Diese Eingabe wird als 4.1.e0 angenommen und es wird eine Fehlermeldung angezeigt.

Fehler bei Fliesskommazahlen und Fliesskomma-Operationen

Bei einer CPU 352 entsteht ein Überlauf, wenn eine REAL-Funktion eine Zahl erzeugt, die größer als 3.402823E+38 oder kleiner als -3.402823E+38 ist. Bei allen anderen 90-30 Modellen, die Fliesskomma-Operationen unterstützen, ist der Bereich größer als 2^{16} oder kleiner als $-2^{16.}$ Liegt Ihre Zahl außerhalb des angegebenen Bereichs, wird der OK-Ausgang der Funktion abgeschaltet und das Ergebnis auf plus unendlich (bei einer Zahl größer als 3.402823E+38 bei der CPU 352 oder oder 2^{16} bei allen anderen Modellen) oder auf minus unendlich (bei einer Zahl kleiner als – 3.402823E+38 oder -2^{16} bei allen anderen Modellen) eingestellt. Durch Abfrage des OK-Ausgangs können Sie feststellen, wo dies stattfindet.

POS_INF	= 7F800000h	– IEEE-Hexadezimaldarstellung von "plus unendlich"
NEG_INF	= FF800000h	– IEEE-Hexadezimaldarstellung von "minus unendlich"

Hinweis

Verwenden Sie Software-Fliesskommazahlen (alle Modelle mit Fliesskommaarithmetik, mit Ausnahme der CPU 352), werden die Zahlen bei ±1.175494E–38 auf Null (0) gerundet.

Werden die durch Überlauf erzeugten Unendlichwerte als Operanden für andere REAL-Funktionen verwenden, kann es dadurch zu einem undefinierten Ergebnis kommen. Wir nennen solch ein undefiniertes Ergebnis "keine Zahl" (NaN). Das Ergebnis der Addition von plus unendlich und minus unendlich ist zum Beispiel undefiniert. Wird die Funktion ADD_REAL aufgerufen mit den Operanden plus unendlich und minus unendlich, dann ist das Ergebnis "NaN".

Bei einer CPU 352 erzeugt jede Funktion, die ein NaN hervorbringen kann, ein spezielles NaN, das die Funktion kennzeichnet:

NaN_SW	= FFFFFFFFh	– Software-Fliesskomma NaN
NaN_ADD.	= 7F81FFFFh	– Hexadezimaler Fehlerwert bei REAL-Addition.
NaN_SUB	= 7F81FFFFh	– Hexadezimaler Fehlerwert bei REAL-Subtraktion.
NaN_MUL	= 7F82FFFFh	– Hexadezimaler Fehlerwert bei REAL-Multiplikation.
NaN_DIV	= 7F83FFFFh	– Hexadezimaler Fehlerwert bei REAL-Division.
NaN_SQRT	= 7F84FFFFh	– Hexadezimaler Fehlerwert bei REAL-Wurzelziehen.
NaN_LOG	= 7F85FFFFh	– Hexadezimaler Fehlerwert bei REAL-Logarithmus.
NaN_POW0	= 7F86FFFFh	– Hexadezimaler Fehlerwert bei REAL-Exponent.
NaN_SIN	= 7F87FFFFh	– Hexadezimaler Fehlerwert bei REAL-Sinus.
NaN_COS	= 7F88FFFFh	– Hexadezimaler Fehlerwert bei REAL-Kosinus.
NaN_TAN	= 7F89FFFFh	– Hexadezimaler Fehlerwert bei REAL-Tangens.
NaN_ASIN	= 7F8AFFFFh	– Hexadezimaler Fehlerwert bei REAL-Arkussinus.
NaN_ACOS	= 7F8BFFFFh	– Hexadezimaler Fehlerwert bei REAL-Arkuskosinus.
NaN_BCD	= 7F8CFFFFh	– Hexadezimaler Fehlerwert bei Umwandlung BCD in REAL.
REAL_INDEF	= FFC00000h	– REAL-unendlich; Fehler aus Division 0 durch 0.

Alle anderen CPUs, die Fliesskommazahloperationen unterstützen, erzeugen einen NaN-Ausgang: FFFF FFFF.

Wird ein NaN in eine andere Funktion eingespeist, gibt es das Ergebnis weiter. Ist, zum Beispiel, ein NaN_ADD der erste Operand für die Funktion SUB_REAL, ist das Ergebnis von SUB_REAL ein NaN_ADD. Sind beide Operationen einer Funktion NaNs, gibt der erste Operand weiter. Durch diese Eigenschaft, NaNs durch Funktionen weiterzugeben, können Sie die Funktion identifizieren, aus der das NaN herrührt.

Hinweis

Bei einem NaN ist der OK-Ausgang AUS (nicht durchgeschaltet).

Die nachstehende Tabelle erläutert wie der Stromfluss bei Zahlen durchgeschaltet wird, die bei binären Operationen (z.B. Addition, Multiplikation, usw.) als unendlich angezeigt werden. Wie zuvor gezeigt, werden Ausgänge, die die positiven oder negativen Grenzwerte überschreiten, als POS_INF bzw. NEG_INF angezeigt.

Tabelle E-1. Allgemeiner Stromfluss bei Fliesskomma-Operationen

Arbeitsweise	Eingang 1	Eingang 2	Ausgang	Stromfluss
Alle	Nummer	Nummer	Plus oder minus unendlich	Nein
Alle außer Division	Unendlich	Nummer	Unendlich	Ja
Alle	Nummer	Unendlich	Unendlich	Ja
Division	Unendlich	Nummer	Unendlich	Nein
Alle	Nummer	Nummer	NaN	Nein